

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Domen Kržišnik

# **Avtonomni robotski pometač**

DIPLOMSKO DELO  
NA UNIVERZITETNEM ŠTUDIJU

MENTOR: doc. dr. Danijel Skočaj

Ljubljana 2015



To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani [creativecommons.si](http://creativecommons.si) ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

*Besedilo je oblikovano z urejevalnikom besedil L<sup>A</sup>T<sub>E</sub>X.*





Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V zadnjem obdobju se pojavlja vedno več avtonomnih robotov, ki opravljajo različna hišna opravila, kot so avtonomni robotski sesalci in kosilnice. V diplomski nalogi zasnujte ter razvijte avtonomni robotski pometlač, ki bo zmožen samostojno pomesti dvorišče. Za osnovo uporabite navaden ročni pometlač ter ga opremite z ustreznimi motorji, senzorji ter krmilno enoto. Implementirajte primerni krmilni algoritem, ki bo omogočal robustno in učinkovito samostojno delovanje robota. Razviti sistem tudi ustrezno evalvirajte na testnem dvorišču.



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Domen Kržišnik, z vpisno številko **63070122**, sem avtor diplomskega dela z naslovom:

*Avtonomni robotski pometač*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Danijela Skočaja,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 1. februar 2015

Podpis avtorja:



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Pregled sorodnih rešitev</b>	<b>5</b>
2.1	Robotski sesalniki za notranjo uporabo . . . . .	6
2.1.1	Standardni nabor senzorjev . . . . .	8
2.1.2	iRobot Roomba . . . . .	13
2.1.3	Samsung NaviBot . . . . .	18
2.2	Robotske kosilnice . . . . .	21
2.2.1	Standardni nabor senzorjev . . . . .	23
2.2.2	Husqvarna Automower . . . . .	26
2.3	Rezultat primerjave robotskih sesalnikov in kosilnic . . . . .	30
<b>3</b>	<b>Komponente robota</b>	<b>31</b>
3.1	Oris sistema . . . . .	32
3.2	Osnovno ogrodje . . . . .	34
3.3	Pogonski elektromotorji in krmilno vezje . . . . .	38
3.4	Napajanje sistema . . . . .	42
3.5	Odometrija . . . . .	44
3.6	Fizični odbijač . . . . .	46
3.7	Ultrazvočni senzorji . . . . .	48

## KAZALO

3.8	Senzor zakopane žice . . . . .	54
3.9	Komunikacijski sistem . . . . .	59
3.10	Glavno krmilno vezje . . . . .	60
<b>4</b>	<b>Krmilna logika robota</b>	<b>67</b>
4.1	Razvojno okolje . . . . .	68
4.2	Premikanje robota . . . . .	72
4.3	Detekcija ovir . . . . .	76
4.4	Obravnava ovir . . . . .	80
4.5	Naključno premikanje . . . . .	83
4.6	Oddaljen nadzor preko vmesnika bluetooth . . . . .	85
<b>5</b>	<b>Analiza delovanja robota</b>	<b>87</b>
5.1	Testno dvorišče . . . . .	90
5.2	Analiza obiskanosti območja delovanja . . . . .	94
5.3	Analiza zaznavanja ovir . . . . .	98
5.4	Analiza daljinskega upravljalnika . . . . .	106
<b>6</b>	<b>Zaključek</b>	<b>109</b>
6.1	Predlog nadaljnjih izboljšav . . . . .	110

# Povzetek

Na trgu se pojavlja veliko vrst robotov za osebno uporabo, ki opravljajo različne naloge, vendar med njimi nismo zasledili primera komercialno dostopnega robota, ki bi bil sposoben učinkovito počistiti oziroma pomesti zunanje dvorišče. V diplomskem delu smo predstavili rezultat načrtovanja, gradnje ter programiranja avtonomnega robota, ki je sposoben opraviti omenjeno nalogo. Diplomsko delo obsega analizo podobnih robotov, kot so robotski sesalniki in kosilnice. Na podlagi izdelane analize smo pripravili načrt zgradbe robota ter razvoja ustrezne krmilne logike robota za opravljanje omenjene naloge. Zaradi lažjega sestavljanja robota smo za osnovo vzeli obstoječi ročni pometlač podjetja Kärcher, ki smo ga nato nadgradili z električnim pogonskim in krmilnim sistemom ter različnimi senzorji. Pri razvoju robota smo uporabili tudi tehniko 3D tiskanja ter platformo Arduino za razvoj krmilne logike robota. Diplomsko delo vsebuje tudi opis postopka in rezultatov analize delovanja robota na testnem dvorišču. Analizirali smo odstotek pokritosti testnega dvorišča, uspešnost zaznavanja različnih tipov ovir ter odzivnost daljinskega upravljalnika.

**Ključne besede:** avtonomni robotski pometlač dvorišča, avtomatizacija, robotika, roboti za osebno uporabo, 3D tiskanje, Arduino





# Abstract

There is already a wide range of personal/domestic robots on the market capable of performing various tasks. We haven't however been able to find any commercially available robots designed for effectively performing the task of backyard sweeping. This thesis presents the process and end result of planning, assembly and programming of an autonomous robot, capable of performing the above mentioned task. We first analyze robots with similar functions, including robotic vacuum cleaners and lawn mowers in order to define design for the physical structure and the appropriate control logic for the robot, so that it will be able to carry out our task. In order to facilitate physical construction, we used an existing manual sweeper by the company Kärcher as our base platform, which was then upgraded with an electric propulsion, control system and various sensors. During development we took advantage of new technologies, such as 3D printing for physical construction and Arduino platform for control logic of the robot. The thesis also includes analytical process of the robot testing, performed in the backyard under realistic conditions. We analyzed the area of coverage, success rate of different type of obstacles detection and responsiveness of the remote control.

**Keywords:** autonomous robotic backyard sweeper, automation, robotics, robots for personal use, 3D printing, Arduino



# Poglavje 1

## Uvod

Da je robotika nepogrešljiva v različnih industrijah oziroma tovarnah je že dolgo znano. Prvi industrijski robot je bil nameščen v tovarni avtomobilov General Motors že daljnega leta 1961 [1]. Dandanes oziroma v zadnjem desetletju pa postajajo vse bolj priljubljeni tudi roboti za osebno uporabo oziroma roboti, ki so namenjeni lažšanju nekaterih nalog v vsakdanjem življenju posameznika. Najbolj znan primer tega so robotski sesalniki, katerih glavna funkcija je samodejno sesanje poljubnega prostora v stanovanju, ki nadomešča potrebo po ročnem sesanju. Poleg robotskih sesalnikov obstajajo tudi robotske kosilnice, ki so sposobne samodejno pokositi travo, robotski čistilci bazenov, robotski pomivalci tal, robotski čistilniki žlebov itd. Obstaja torej že kar nekaj avtomatiziranih robotov, ki so sposobni opravljati različne naloge, s katerimi se povprečen človek srečuje v svojem vsakdanjiku. Kljub temu da obstaja že ogromno različnih robotov, ki opravljajo različne naloge, pa na dan pisanja diplomskega dela nismo uspeli najti komercialno dostopnega robota, ki bi bil sposoben pomesti oziroma počistiti zunanje dvorišče.

Glavni namen tega diplomskega dela je torej bil sestaviti ter ustrezno programirati robota, ki bo sposoben opraviti nalogo čiščenja oziroma pometanja zunanjega dvorišča. Samo sestavo robota smo si nekoliko olajšali tako, da smo za osnovo vzeli obstoječi ročni pometal dvorišča podjetja Kärcher, ki vsebuje osnovno ogrodje, krtače in zbiralnik za smeti. To osnovo smo nato nadgradili

z električnim pogonskim sistemom, napajalnim sistemom, različnimi senzorji in s krmilnim sistemom. V tem diplomskem delu se bomo omejili le na eno testno dvorišče, vendar pa bi moral robot podobno učinkovito delovati tudi na poljubnem dvorišču. Slika 1.1 ponazarja omejeno testno dvorišče ter robota oziroma končni izdelek.



Slika 1.1: Robotski pometać na testnem dvorišču.

Naše diplomsko delo bomo razdelili na štiri dele oziroma poglavja. V prvem delu bomo najprej raziskali podobne robote, ki rešujejo podobne probleme. Analizirali bomo predvsem robotske sesalnice ter robotske kosilnice, saj lahko potegnemo marsikatero vzporednico z našo nalogo. Malce bolj natančno si bomo pogledali, kako so ti roboti zgrajeni, katere senzorje uporabljajo za kakšen namen, ter kakšne imajo algoritme premikanja oziroma preiskovanja prostora. Na podlagi te analize oziroma ugotovitev smo nare-

dili načrt, kakšna kombinacija senzorjev ter algoritmov premikanja je najbolj optimalna v našem primeru. V drugem poglavju bomo nato predstavili vse te uporabljene senzorje oziroma komponente našega robota, kako delujejo ter kako smo jih integrirali v našo rešitev. Pri namestitvi komponent ter senzorjev na omejeno osnovo robota smo se tudi poslužili relativno nove tehnike 3D tiskanja različnih plastičnih komponent (na podlagi narisane 3D računalniškega modela). Naslednji korak oziroma tretje poglavje bo vsebovalo bolj natančen opis krmilne logike robota, ki zna pravilno interpretirati ter povezati skupaj podatke iz različnih senzorjev, ter nato na podlagi teh informacij sprejemati različne odločitve. Za razvoj oziroma implementacijo krmilne logike smo uporabili platformo Arduino. V zadnjem poglavju diplomskega dela bomo opisali opravljeno testiranje ter analizo delovanja robota na testnem dvorišču. Analizirali smo obiskanost območja delovanja, testirali uspešnost zaznavanja različnih tipov ovir, ter preverili odzivnost daljinskega upravljalnika.



## Poglavje 2

### Pregled sorodnih rešitev

Ker sta si zunanje in notranje okolje (zaprt prostor) med seboj precej različna, smo se v tem poglavju osredotočili najprej na pregled obstoječih rešitev, ki so namenjeni notranji uporabi, ter nato še pregledu obstoječih rešitev za zunanje okolje. V obeh primerih smo se osredotočili samo na robote, ki so komercialno dostopni ter masovno proizvajani.

Razlog, zakaj smo se v tem poglavju osredotočili samo na komercialno dostopne oziroma masovno proizvajane robote, je predvsem v tem, da je naš cilj zgraditi robota iz cenovno dostopnih komponent/senzorjev. Zato smo v tem poglavju naredili analizo, kako so zgrajeni ter kako delujejo roboti, ki so namenjeni široki potrošnji oziroma so cenovno dostopni, kar pomeni, da uporabljajo tudi cenovno dostopne komponente oziroma senzorje. Za pregled rešitev, namenjenih notranji uporabi smo se osredotočili na robotske sesalnike, za pregled rešitev, namenjenih zunanji uporabi, pa na robotske kosilnice.

## 2.1 Robotski sesalniki za notranjo uporabo

Prvi komercialno dostopni robotski sesalnik je bil Electrolux Trilobite, ki je bil prvič predstavljen leta 1997 [2]. Dandanes pa na svetovnem tržišču obstaja že ogromno različnih cenovno ugodnih robotskih avtonomnih sesalnikov. Med seboj se razlikujejo predvsem v kvaliteti izdelave, številu in kvaliteti uporabljenih senzorjev ter samem algoritmu delovanja oziroma preiskovanju prostora. V grobem jih glede na način preiskovanja prostora lahko delimo na dve skupini:

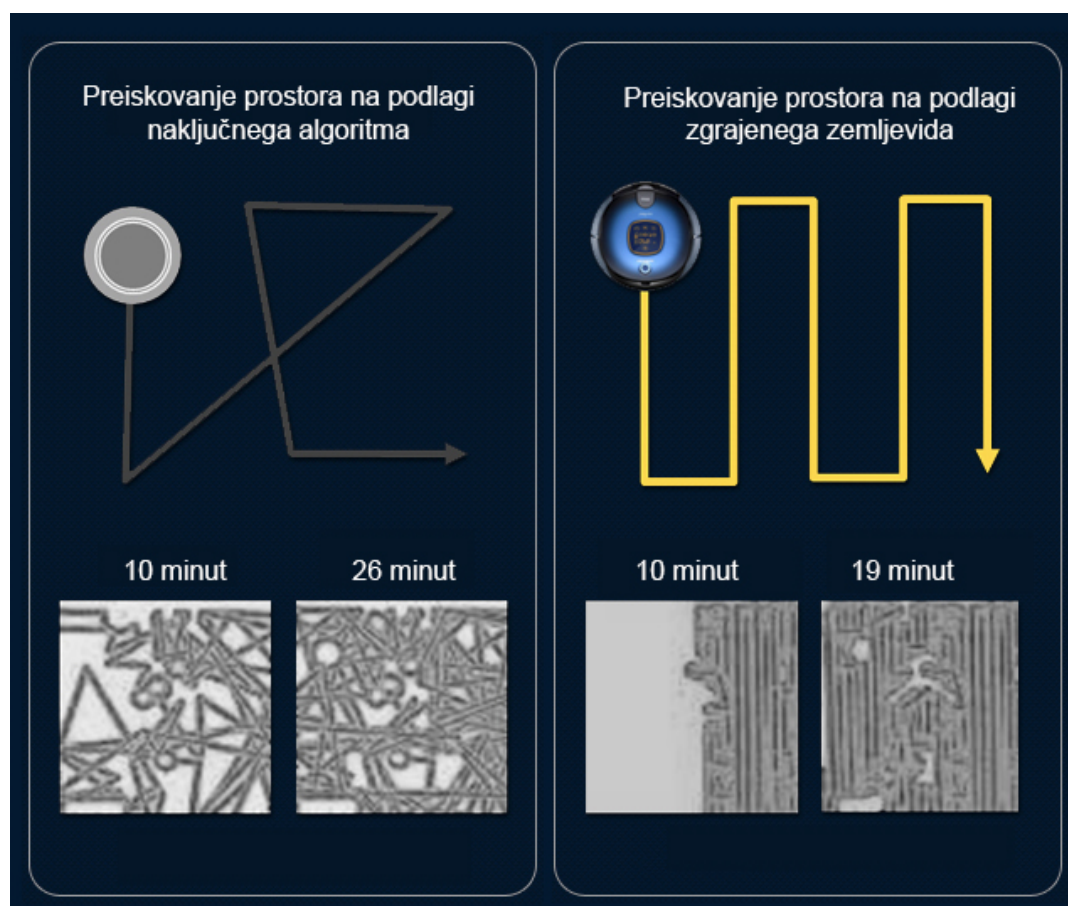
1. Roboti, ki preiskujejo prostor na osnovi naključnega algoritma.
2. Roboti, ki prostor preiskujejo na podlagi zgrajenega zemljevida.

Kot primer preiskovanja prostora na osnovi naključnega algoritma lahko štejemo robotske sesalnike proizvajalca iRobot Roomba, ki poskušajo principe naključnega premikanja čim bolj optimizirati na podlagi omejenega nabora senzorjev. Kot primer preiskovanja prostora na podlagi zgrajenega zemljevida lahko štejemo robotske sesalnike proizvajalca Samsung NaviBot. Ti roboti v grobem delujejo tako, da imajo vgrajeno kamero, ki je usmerjena proti stropu ter periodično zajema slike stropa sobe. Nato pa na osnovi obdelave teh slik robot poskuša zgraditi nek približen zemljevid oziroma začrta pot, po kateri se bo poskušal premikati. Slika 2.1 predstavlja primerjavo omenjenih principov preiskovanja prostora.

V osnovi sta obe omejeni skupini robotov močno prilagojeni uporabi v notranjem prostoru. To pomeni, da sam algoritem premikanja ter nabor senzorjev upoštevajo določene predpostavke. Nekaj pomembnejših predpostavk je:

1. Robot deluje v zaprtem oziroma omejenem prostoru, kar v bistvu pomeni, da bo načeloma tudi z uporabo popolnoma naključnega vzorca premikanja tak robot nekoč preiskal oziroma pokrtil celotni prostor.





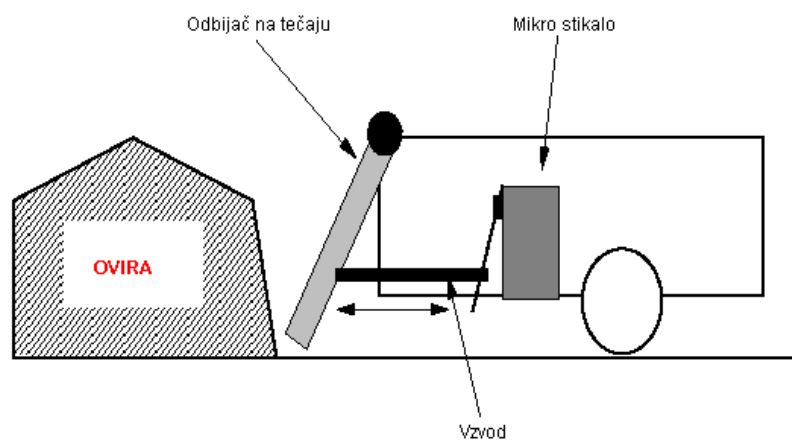
Slika 2.1: Primerjava preiskovanje prostora na podlagi zemljevida in na osnovi naključnega algoritma [13].

2. Robot deluje na ravni in vodoravni površini, ki je relativno enakomerna z le nekoliko manjšimi višinskimi razlikami, kot so na primer pragi in preproge. Ta predpostavka omogoča proizvajalcem, da poenostavijo pogonsko tehniko robota z uporabo sistema le dveh pogonskih koles, ki so specifično načrtovana za tako podlago.
3. Robot deluje v prostoru, ki ima v povprečju majhno neposredno izpostavljenost soncu. To pomeni, da je v samem prostoru relativno majhen delež sončnih infrardečih žarkov, kar v bistvu omogoča proizvajalcem uporabo nizkocenovnih infrardečih senzorjev za merjenje oddaljenosti objektov od robota. Uporaba takih senzorjev v zunanjem okolju oziroma na neposredni izpostavljenosti soncu je skoraj popolnoma neuporabna.
4. V povprečju so predmeti, ki jih mora robot posesati, relativno majhni oziroma je predpostavka, da bo robot v večini primerov sesal prah, živalsko dlako, pesek ter ostale podobne majhne delce. To omogoča proizvajalcem, da se omejijo na relativno majhen zbiralnik za smeti in s tem zmanjšajo potrebno fizično velikost robota, kar posledično pomeni, da lahko robot doseže bolj prostorsko omejene koticke.

### 2.1.1 Standardni nabor senzorjev

Če med seboj primerjamo cenovno ugodne robotske sesalnike različnih proizvajalcev, lahko ugotovimo, da obstaja kar nekaj podobnosti v naboru osnovnih senzorjev. Spodaj se nahaja seznam najpogostejše uporabljenih senzorjev ter grob opis njihovih funkcij oziroma namembnosti:

1. **Fizični odbijač** – uporablja se predvsem za detekcijo ovir oziroma za detekcijo trka. Običajno temelji na principu vzvoda in mikrostikal, ki se sprožijo ob določeni sili pritiska na odbijač. Slika 2.2 prikazuje primer fizičnega odbijača.



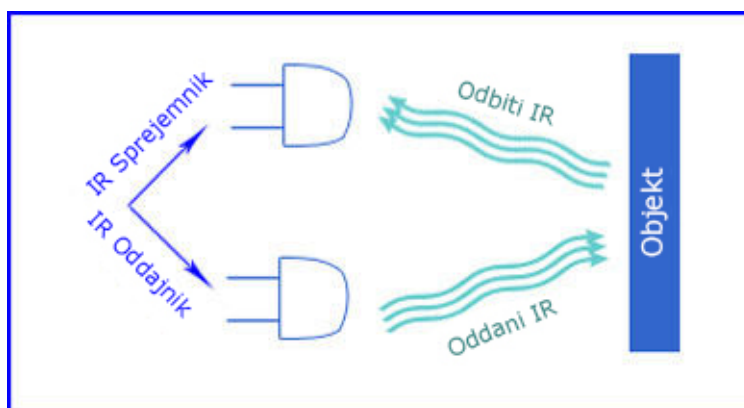
Slika 2.2: Primer delovanja fizičnega odbijača.

2. Infrardeči senzor za detekcijo previsov – običajno sta to dva ali več infrardečih senzorjev na sprednji strani robota, ki so usmerjeni navzdol proti tlom. Algoritem robota pričakuje, da se bo signal infrardečega senzorja takoj odbil od tal nazaj v sprejemnik in če temu ni tako, potem sklepa, da je prišel do npr. roba stopnic ali kakšnega drugega podobnega previsa. Slika 2.3 prikazuje primer delovanja infrardečega senzorja za detekcijo previsov.



Slika 2.3: Delovanje infrardečega senzorja za detekcijo previsov [12].

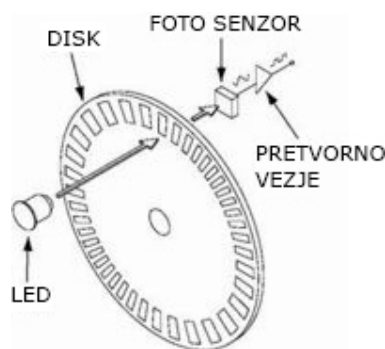
3. **Infrardeči senzor za detekcijo ovir** – običajno je to en ali več infrardečih senzorjev, ki so nameščeni na odbijač ter usmerjeni naravnost naprej oziroma vzporedno s tlemi. Glavni namen teh senzorjev je običajno detekcija ovir, da preprečijo pretirano zaletavanje robota v te ovire. Prav tako se velikokrat uporabljajo tudi za detekcijo sten oziroma omogočajo robotu, da se premika vzporedno čisto zraven stene oziroma ostalih podobnih objektov, ter tako poskuša čim bolj počistiti umazanijo ob robovih sobe. Infrardeči senzorji prav tako omogočajo tudi merjenje dejanske razdalje do objektov oziroma ovir, kar potem omogoča robotu, da zmanjša hitrost premikanja, če zazna nek objekt pred sabo in se s precej zmanjšano hitrostjo poskuša čim bolj približati objektu, dokler se ga skoraj ne dotakne. To v bistvu omogoča precej izboljšano pokritost prostora. Slika 2.4 prikazuje primer delovanja infrardečega senzorja za detekcijo ovir.



Slika 2.4: Delovanje infrardečega senzorja za detekcijo ovir [10].

4. **Akustični senzor za merjenje količine umazanije** – to je običajno en akustični senzor, ki meri količino vibracij na kovinski ploščici, ki jih povzročajo udarci umazanije, ki jo pobere robot. Večja kot je količina umazanije, ki se zaleti v ploščico, več je vibracij in tako posledično algoritem ugotovi, da je ta specifični del prostora bolj umazan in se poskuša robot več časa zadrževati na tem odseku.

5. **Senzor za odometrijo lege robota** – običajno so ti senzorji nameščeni neposredno na posamezen električen motor ter merijo njegovo hitrost vrtenja. Merjenje hitrosti poteka s pomočjo optičnega senzorja, in sicer tako, da je običajno kar na pogonsko gred motorja nameščen kovinski disk, ki ima na obodu enakomerno razporejene luknje. Na eni strani diska je nepremično pritrjena IR LED dioda, na drugi strani diska pa je nepremično pritrjen optični senzor, ki zazna prisotnost IR svetlobe. Sam disk se vrti skupaj z osjo motorja, medtem ko sta senzor in dioda statična. Tako lahko posledično merimo kolikokrat na sekundo optični senzor zazna svetlobo in tako lahko nato izračunamo samo hitrost motorja [11]. Slika 2.5 ponazarja opisano delovanje senzorja.



Slika 2.5: Delovanje odometričnega senzorja [11].

6. **Infrardeči sprejemnik za virtualne zidove in bazno postajo** – predstavlja optični infrardeči senzor oziroma sprejemnik, ki zna zaznati infrardečo svetlobo. Običajno predstavlja ta senzor najvišjo točko robota. Uporablja se pa predvsem zato, da robot najde pot nazaj do svoje bazne oziroma polnilne postaje ter hkrati tudi za detekcijo virtualnih zidov. Virtualni zid je ločena enota, ki vsebuje infrardeči oddajnik, ki oddaja svetlobo v ravni liniji. Uporablja se zato, da lahko robotu postavimo nek virtualni zid oziroma določimo mejo, preko katere ne želimo, da se premakne. Bazna postaja deluje tako, da ima nameščen infrardeči oddajnik, ki seva infrardečo svetlobo v približno 45 stopinj

širokem snopu v smeri polnilnih kontaktov. Ko robot preklopi v način iskanja bazne postaje, sam poskuša locirati ta snop infrardeče svetlobe ter mu slediti do njegovega izvora oziroma dokler ne zazna polnilnih kontaktov [3]. Slika 2.6 ponazarja opisano delovanje mehanizma.



Slika 2.6: Prikaz delovanja bazne postaje.

7. **Senzor za detekcijo tal** – uporablja se predvsem zato, da robot zazna, kdaj kolesa nimajo več stika s tlemi oziroma da je nekdo robota dvignil v zrak. Uporaben je zato, da robot ve, kdaj naj ustavi motorje oziroma resitira svoj algoritem čiščenja. Senzor je običajno implementiran tako, da ima posamezno kolo nameščeno na posebno nihalo z nameščenim končnim stikalom, ki je sproženo samo v primeru, ko je kolo v zraku oziroma nima stika s tlemi.
8. **Senzor stanja ter polnjenja baterije** – namensko vezje, ki zna upravljati s polnjenjem in preveriti stanje baterije.

Ta nabor senzorjev predstavlja standardni nabor senzorjev v svetu robotskih sesalnikov in ga v določenih variacijah oziroma izvedbah uporablja večina proizvajalcev robotskih sesalnikov.

### 2.1.2 iRobot Roomba

Podjetje iRobot je bilo ustanovljeno leta 1990 na Massachusetts inštitutu za tehnologijo (MIT) v Cambridgeu, ZDA. Podjetje izdeluje več različnih robotov, vendar pa so njihova najmočnejša linija prav robotski sesalci. Poleg robotskih sesalnikov podjetje izdeluje še robote za pomivanje tal, robote za čiščenje bazenov ter robote za čiščenje žlebov [4]. Njihova linija robotskih sesalnikov, imenovana iRobot Roomba, vsebuje več različnih modelov, ki se med seboj razlikujejo predvsem po funkcionalnostih, ki jih ponujajo.

Ta pregled zajema analizo nekaj najpomembnejših funkcionalnosti, na podlagi katerih smo potem razvili našo implementacijo premikanja robota. Vsi modeli linije Roomba so predstavniki prej opisane skupine robotov, ki preiskujejo prostor na osnovi naključnega algoritma. Slika 2.7 predstavlja primer izdelka omejenega podjetja.



Slika 2.7: Primer robotskega sesalnika iRobot Roomba 560 [7].

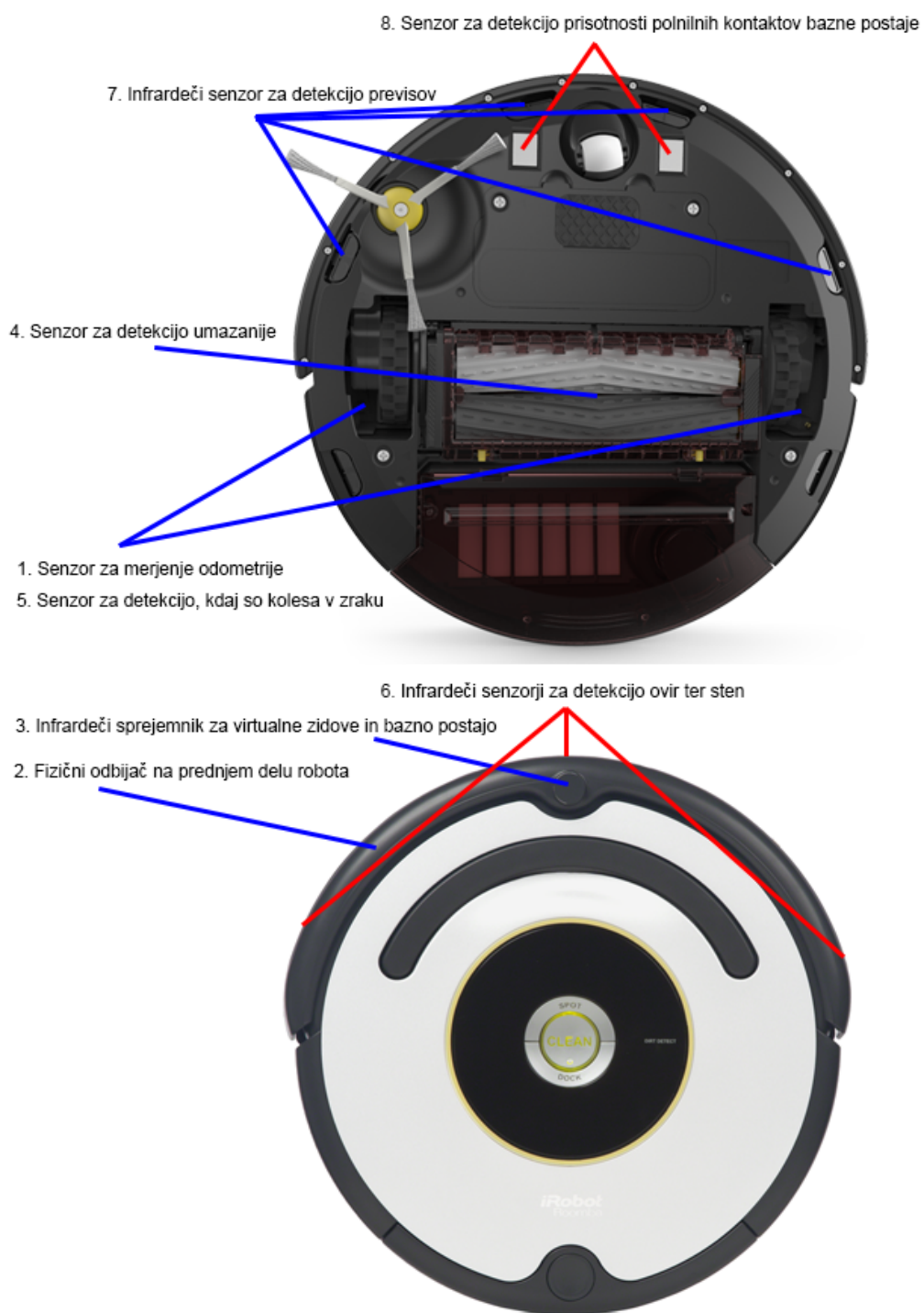
Roomba uporablja precej standardni nabor senzorjev, ki vključuje [6]:

1. Senzor za merjenje odometrije.
2. Fizični odbijač na prednjem delu robota.
3. Infrardeči sprejemnik za detekcijo virtualnih zidov in bazno postajo.
4. Senzor za detekcijo umazanije.
5. Senzor za detekcijo tal oziroma detekcijo, kdaj je robot v zraku.
6. Infrardeči senzorji za detekcijo ovir ter sten.
7. Infrardeči senzor za detekcijo previsov.
8. Senzor za detekcijo prisotnosti polnilnih kontaktov bazne postaje.
9. Senzor stanja ter polnjenja baterije.

Kot smo že v prejšnjem poglavju omenili, predstavlja ta nabor senzorjev standardni nabor v svetu robotskih sesalnikov. Slika 2.8 predstavlja vizualno identifikacijo posameznega senzorja. Roomba za izvedbo določenega algoritma uporablja določeno kombinacijo senzorjev na različne načine. Podrobnejši opis posameznih senzorjev oziroma kako delujejo ter za kaj se uporabljajo smo zapisali v prejšnjem poglavju.

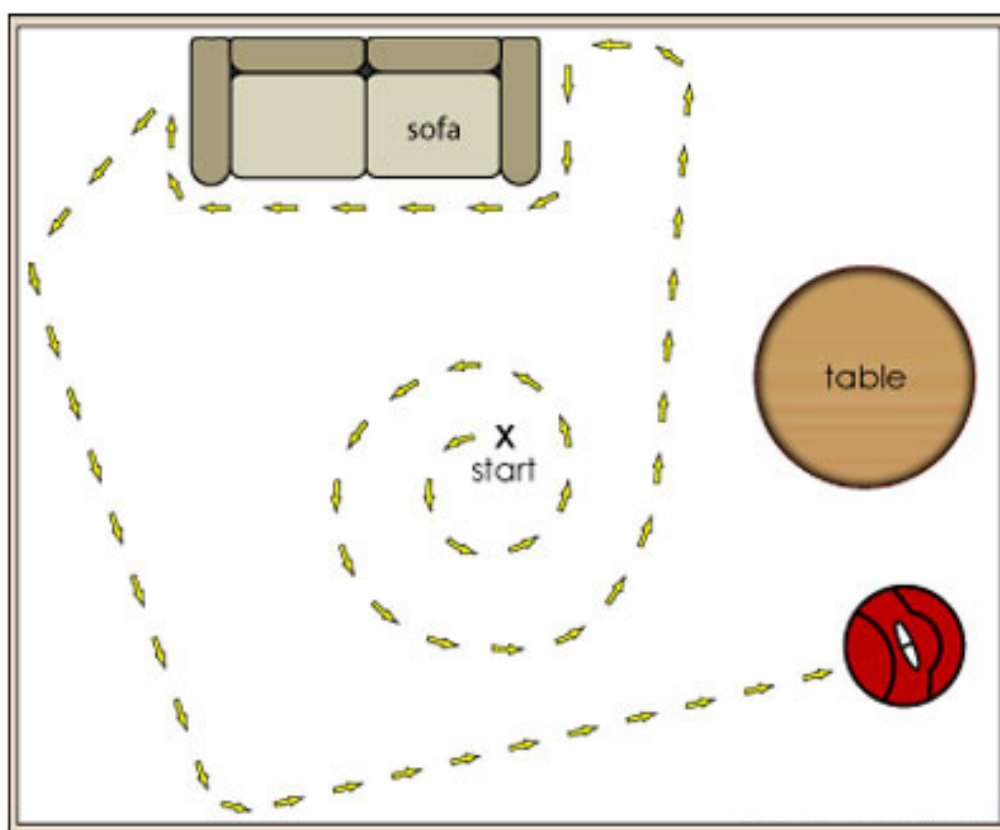
Prvi algoritem, ki ga iRobot Roomba izvede, ko pritisnemo na tipko “Clean”, je, da poskuša izračunati velikost prostora in nato na podlagi te informacije določi, koliko časa bo izvajal “čistilni” cikel. Velikost prostora poskuša izračunati tako, da pošlje infrardeči signal iz zgornjega oddajnika ter nato na podlagi časa, ki ga signal potrebuje, da pride nazaj do sprejemnika, poskuša izračunati približno velikost sobe. Robot nato preide v “čistilni” cikel, ki je v bistvu nekoliko optimizirano naključno premikanje za določeno časovno obdobje. Pri čemer robot predpostavi, da se nahaja približno na sredini sobe in najprej naredi začetno pot v obliki spirale, dolge približno ena in pol obrata, ter nato nadaljuje naravnost naprej, dokler ne zazna prve





Slika 2.8: Senzorji robotskega sesalnika iRobot Roomba [7].

ovire. Ko robot zazna prvo oviro, sklepa, da je prišel do roba prostora in poskuša slediti robu te ovire dokler ne zazna nove. Ob zaznavi nove ovire poskuša počistiti okoli nje ter poiskati prvo ravno linijo na drugi konec prostora. Ko najde to linijo, pa poskuša delati čim daljše ravne poti iz enega konca prostora do drugega, dokler se ne zaključi "čistilni" cikel oziroma dokler ne poteče na začetku definiran čas čiščenja. Po tem pretečenem času robot sklepa, da je zadostno počistil celotni prostor. Slika 2.9 ponazarja slikovno predstavitev opisanega algoritma premikanja robota iRobot Roomba (algoritem je bil povzet po [5]).



Slika 2.9: Vizualizacija algoritma premikanja robota iRobot Roomba [5].

Robot vedno, ko se premika, spremlja vse opisane senzorje in se nanje ustrezno odziva. Za detekcijo ovir uporablja infrardeče senzorje na sprednji strani (senzor številka 6) ter fizični odbijač (senzor številka 2). Če robot preko enega izmed teh dveh senzorjev zazna, da se je zaletel v oviro, se vedno začne obračati v levo smer toliko časa, dokler se ne izogne oviri oziroma je ne zazna več s senzorji [6]. Ker je robot okrogle oblike ter ima kolesa točno na sredini, se lahko vedno obrne za poljuben kot brez nevarnosti, da bi se zataknil za kakšen predmet. Prav tako robot vedno spremlja prednje infrardeče senzorje za detekcijo previsov (senzor številka 7), zato da zazna, kdaj je prišel na primer do kakšni stopnic, balkona itd. Če to zazna, obravnava situacijo na enak način kot v prejšnjem odstavku, kjer smo opisovali odziv na ovire [9]. Enak odziv ima tudi, če preko infrardečega sprejemnika za virtualne zide (senzor številka 3) zazna infrardeči signal. V celotnem "čistilnem" ciklu robot ves čas spremlja senzor za detekcijo umazanije (senzor števila 4). Če zazna, da je določen odsek prostora bolj umazan kot ostali, naredi na tisti lokaciji spiralno pot z zmanjšano hitrostjo oziroma posveti tisti lokaciji nekoliko več časa [8].

Če robot preko senzorja stanja baterije (senzor številka 9) zazna, da je baterija že skoraj izpraznjena, poskuša poiskati svojo bazno postajo. To naredi tako, da preklopi v stanje iskanja bazne postaje, kar pomeni, da zaradi varčevanja z baterijo izklopi sesalnik ter se začne premikati v ravnih linijah ter na podlagi infrardečega sprejemnika na sprednjem odbijaču (senzor številka 3) poskuša poiskati infrardeči signal bazne postaje. Ko robot ta signal uspešno najde, mu poskuša slediti do izvora oziroma mu poskuša slediti toliko časa, dokler na senzorju za detekcijo prisotnosti polnilnih kontaktov bazne postaje (senzor številka 8) ne zazna prisotnosti električne napetosti. Takrat začne polniti baterijo ter čaka na novo interakcijo s strani uporabnika [3]. Če v kateremkoli ciklu delovanja robot preko senzorjev na pogonskih kolesih (senzor številka 5) zazna, da so kolesa v zraku, takoj prekrine trenutni algoritem oziroma ustavi pogonska kolesa, sproži zvočni signal ter čaka na interakcijo s strani uporabnika. iRobot Roomba za napajanje celotnega sis-

tema uporablja Ni-MH (Nikl-Metal Hydrid) baterijo, ki ob polni kapaciteti zadošča za približno 2 uri čiščenja, preden jo je potrebno ponovno napolniti [7].

### 2.1.3 Samsung NaviBot

Multinacionalno podjetje Samsung je na globalni trg robotskih sesalnikov vstopilo bistveno pozneje kot podjetje iRobot, svojo prvo linijo globalno dostopnih robotskih sesalnikov, imenovano NaviBot so izdali šele leta 2010. Samsung ima tudi zelo močno ter priznano linijo običajnih sesalnikov, na podlagi katere je podjetje pridobilo veliko izkušenj ter znanja, ki je bilo potem uporabljeno pri njihovi liniji robotskih sesalnikov. Največja in najpomembnejša razlika robotov Samsung NaviBot v primerjavi z roboti podjetja iRobot Roomba je, da NaviBot za navigacijo uporablja kamero oziroma princip vizualnega kartiranja prostora (*angl. Visionary mapping system*), kar pomeni, da spada v skupino robotov, ki preiskujejo prostor na podlagi zgrajenega zemljevida, medtem ko vsi roboti podjetja iRobot Roomba preiskujejo prostor na osnovi naključnega algoritma. Slika 2.10 predstavlja primer izdelka podjetja Samsung.



Slika 2.10: Primer robotskega sesalnika Samsung NaviBot S [12].

Kot smo že omenili, robot Samsung NaviBot uporablja za navigacijo po prostoru kamero. Kamera, ki snema s hitrostjo 30 slik na sekundo, se nahaja na prednjem delu robota in je obrnjena navzgor proti stropu. Celoten sistem, ki ga je Samsung poimenoval inteligentni vizualni kartirni plus sistem, (*angl. Intelligent Visionary Mapping Plus system*) deluje tako, da preko kamere zajema slike stropa, jih ustrezno obdelava z določenimi algoritmi ter nato združi te informacije še s podatki iz infrardečih senzorjev (podoben način delovanja kot pri prejšnjem poglavju, kjer smo opisovali robota iRobot Roomba) in si s tem zgradi virtualni zemljevid prostora. Za izvedbo svojega kognitivnega kartiranja prostora uporablja kar dva CPU čipa, saj je obdelava slik v realnem času procesorsko precej zahtevna naloga. Robot nato na podlagi tega virtualnega zemljevida prostora izračuna najbolj optimalno pot premikanja, da bo v čim krajšem času prevozil celotni prostor. Robot zna v vsakem danem trenutku določiti svojo pozicijo v prostoru ter si zapomni, kateri del svoje načrtovane poti je že prevozil in kateri ne. Tako v primeru, da mu je zmanjkalo energije in se je moral vrniti nazaj na bazno postajo, ve kje je že počistil oziroma kje je prej ostal ter nadaljuje čiščenje od tiste točke naprej in ne od začetka. Kar v bistvu omogoča precej krajši čistilni čas in boljšo pokritost v primerjavi z roboti iRobot Roomba. Kot primerjava, robot Samsung NaviBot pokrije 90 % prostora v istem času, kot robot iRobot Roomba pokrije 67 % (algoritem je bil povzet po [13]). Slika 2.11 prikazuje primer izračunane načrtovane poti premikanja robota. Sicer Samsung ne trdi, da je njihov zgrajen virtualni zemljevid vedno 100 % točen, vendar pa v primerjavi z naključnim algoritmom omogoča bistveno bolj inteligentno premikanje oziroma navigacijo po prostoru. Robot si prav tako zapomni lokacije vseh ovir ter neprestano posodablja svoj virtualni zemljevid na podlagi informacij, pridobljenih iz svojih senzorjev.



Slika 2.11: Primer načrtovane poti premikanja robotskega sesalnika Samsung NaviBot S [12].

Poleg kamere ima robot Samsung NaviBot relativno podoben nabor senzorjev kot v prejšnjem poglavju, kjer smo opisovali nabor senzorjev iRobot Roomba, oziroma vsebuje standardni nabor senzorjev (opisano v Poglavju 2.1). Torej popolni seznam senzorjev robota Samsung NaviBot vključuje:

1. Senzor za merjenje odometrije.
2. Fizični odbijač na prednjem delu robota.
3. Infrardeči sprejemnik za detekcijo virtualnih zidov in bazne postaje.
4. Senzor za detekcijo umazanije.
5. Senzor za detekcijo tal oziroma detekcijo, kdaj je robot v zraku.
6. Infrardeči senzorji za detekcijo ovir ter sten.
7. Infrardeči senzor za detekcijo previsov.

8. Senzor za detekcijo prisotnosti polnilnih kontaktov bazne postaje.
9. Senzor stanja ter polnjenja baterije.
10. Kamera za navigacijo.

Robot zazna, kdaj mu bo kmalu zmanjkalo električne energije ter se vrne nazaj na svojo bazno postajo, da se bo napolnil. Za zaznavanje točne lokacije bazne postaje robot Samsung Navibot uporablja infrardeči sprejemnik ter relativno podoben algoritem kot robot iRobot Roomba. S to razliko, da ko enkrat robot najde svojo bazno postajo, si to lokacijo shrani v svoj zgrajeni virtualni zemljevid in tako bo naslednjič, ko se bo moral vrniti nazaj na bazno postajo, precej dobro vedel, kje se bazna postaja nahaja. To robotu Samsung Navibot omogoča, da mu ni potrebno vsakič znova preiskovati prostora, da bi zaznal njeno lokacijo, ampak zna izračunati najbolj optimalno oziroma najhitrejšo pot, da se bo vrnil nazaj. Kar tudi omogoča, da robot bistveno kasneje oziroma z bistveno manj rezervne električne energije preklopi v fazo vračanja nazaj na bazno postajo [12].

Glavni fokus podjetja Samsung pri njihovi liniji robotskih sesalnikov NaviBot je vedno bil, da jih poskušajo narediti čim bolj pametne in z vsako novo verzijo so nekoliko pametnejši. Lahko bi tudi rekli, da je robot Samsung NaviBot S trenutno (v času pisanja te naloge) najbolj napreden oziroma najpametnejši predstavnik masovno proizvedenih robotskih sesalnikov.

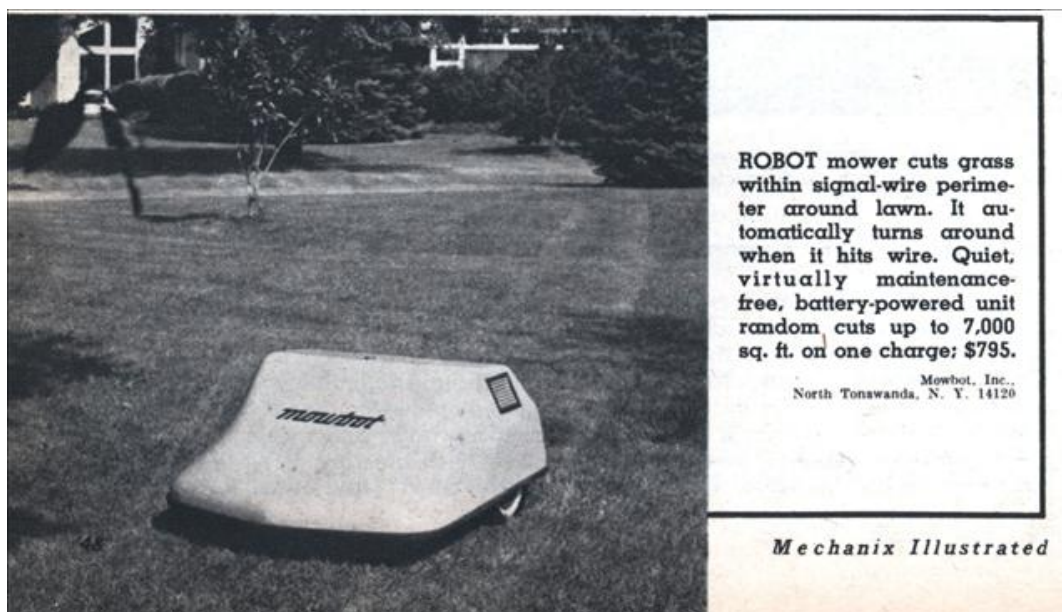
## 2.2 Robotske kosilnice

Robotske kosilnice so roboti, ki so namenjeni uporabi v zunanjem prostoru, v katerem pa kar nekaj izmed predpostavk, ki so bile omenjene v Poglavju 2.1 odpade. Torej, če primerjamo omejene predpostavke glede na zunanji prostor:

1. Robot ne deluje v zaprtem prostoru, zato je potrebno dodati dodatne senzorje, da ga omejimo na specifično področje.

2. Zunanja površina oziroma travnik je zelo redko popolnoma vodoraven, zato je potrebno ustrezno prilagoditi pogonski sistem.
3. Robot je izpostavljen neposredni sončni svetlobi, ki vključuje tudi IR žarke, kar pomeni, da so nizkocenovni infrardeči senzorji skoraj popolnoma neuporabni.
4. Trava, ki jo bo moral robot kositi, je lahko različne velikosti.

Zunanji svet je v osnovi bistveno bolj kompleksen oziroma je potrebno upoštevati veliko več spremenljivk kot v notranjem prostoru. Prva komercialna robotska kosilnica je bila MowBot podjetja MowBot, Inc, ki je bila predstavljena in patentirana že leta 1969 in je vsebovala že kar nekaj funkcij, ki jih lahko tudi opazimo pri današnjih modernih robotih [14]. Slika 2.12 predstavlja izrezek članka iz ameriške revije *Mechanix Illustrated* (izdaja Jan, 1969).



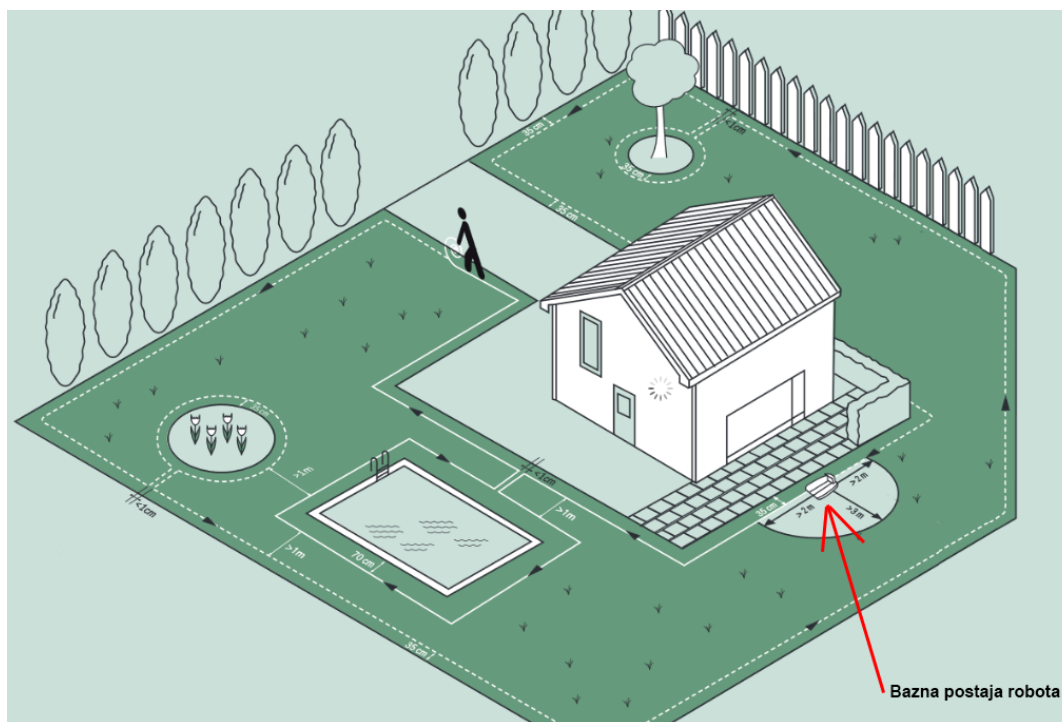
Slika 2.12: Prva robotska kosilnica MowBot [14].



### 2.2.1 Standardni nabor senzorjev

Dandanes je na tržišču na voljo kar nekaj različnih robotskih kosilnic različnih proizvajalcev, vendar je med njimi kar nekaj podobnosti oziroma podobno kot pri robotskih sesalnikih tudi pri kosilnicah obstaja določen nabor senzorjev, ki nekako velja kot standardni set senzorjev v svetu robotskih kosilnic. Ta nabor vključuje:

1. **Zakopana žica za določitev območja delovanja** – ker so lahko zunanja okolja med seboj precej različna oziroma ima vsak uporabnik lahko precej specifične želje, kje želi, da njegov robot pokosi travo, je skoraj obvezen del vsake robotske kosilnice zakopana žica za določitev območja delovanja. To je v bistvu najcenejši način, kako lahko robotu v zunanjem okolju določimo njegovo območje delovanja. Ta princip je že uporabljala prva komercialna robotska kosilnica MowBot leta 1969. Sistem deluje tako, da na robovih območja znotraj katerega želimo, da naša robotska kosilnica pokosi travo, zakopljemo ali položimo žico, ki je povezana v zanko nazaj v bazno postajo. Bazna postaja nato v to žico oddaja električni sinusni signal, kar v bistvu spremeni žico v radijski oddajnik, ki ima zelo majhen domet. Za zaznavanje prisotnosti te zakopane žice pa robot uporablja dva za to namenjena senzorja, običajno nameščena v sprednjem levem in desnem kotu ohišja robota. Ta dva senzorja sta običajno tuljave ter namensko vezje, ki zna zaznati, kdaj je bil v tuljavah induciran električni tok, ki ga je povzročila zakopana žica. Tako robot na podlagi prisotnosti električnega toka v tuljavah ve, kdaj je prišel do roba svojega območja in mora obrniti [16]. Prav tako pa to zakopano žico robot uporablja tudi zato, da zna priti oziroma poiskati svojo bazno postajo. Slika 2.13 prikazuje primer območja, ki je bil določen na podlagi opisanega principa zakopane žice.

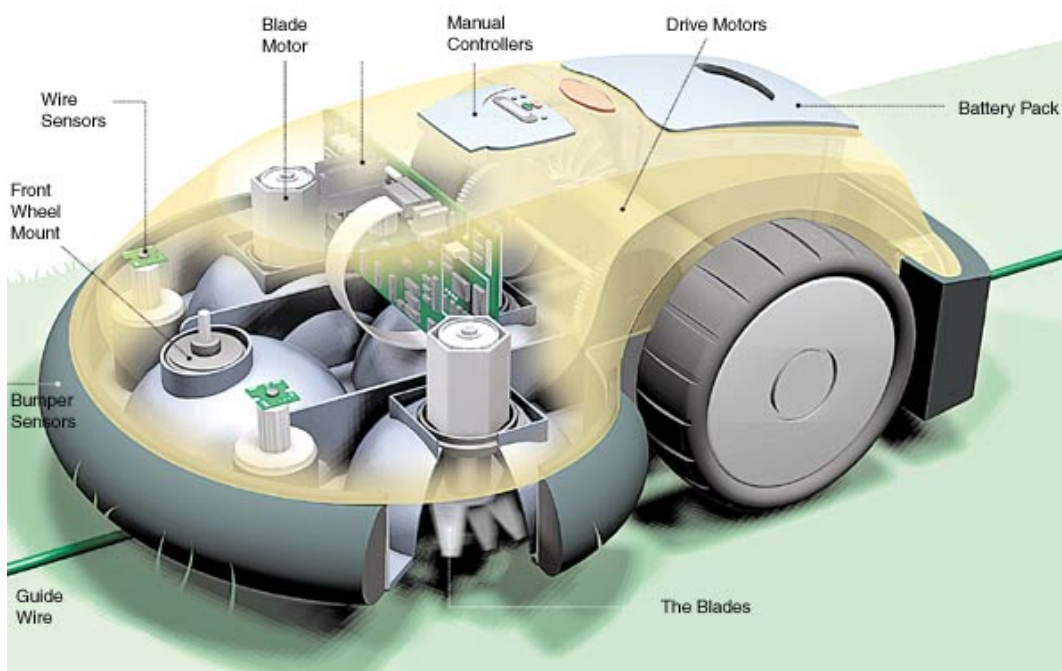


Slika 2.13: Primer določenega območja delovanja robotske kosilnice s pomočjo principa zakopane žice [15].

2. **Fizični odbijač** – podobno kot pri robotskih sesalnikih je tudi pri robotskih kosilnicah fizični odbijač namenjen predvsem zaznavanju, kdaj se je robot zaletel v oviro. V grobem se pri robotskih kosilnicah uporabljata dva različna tipa fizičnih odbijačev. Prvi tip temelji na principu vzvodov in mikrostikal podobno kot pri robotskih sesalnikih oziroma kot je bilo to že opisano v Poglavju 2.1.1. Drugi tip fizičnega odbijača pa temelji na principu zračne blazine ter merjenju razlike pritiska v tej zračni blazini. Namreč, ko se zračna blazina stisne, se poveča pritisk in preko senzorja robot ve, da se je zaletel v oviro.
3. **Senzor za dež** – nekateri modeli robotskih kosilnic vsebujejo tudi senzorje za dež. Uporabljajo se zato, da robot zazna, kdaj je začelo deževati in kdaj se mora vrniti nazaj na svojo bazno postajo, saj so

namreč pogoji na mokri travi po navadi bistveno slabši in lahko robot zaradi slabšega oprijema s podlago tudi nekje obtiči oziroma uniči površino z vrtenjem koles v prazno. Senzor je običajno nameščen na zgornji del robota, deluje pa tako, da vsebuje posebne prevodnike, ki v primeru, da pridejo v stik z vodo, sklenejo tokokrog in tako lahko robot potem posledično zazna, da je začelo deževati [17].

Če primerjamo fizično konstrukcijo oziroma osnovno obliko robotske kosilnice v primerjavi z robotskimi sesalniki, lahko ugotovimo, da je kar nekaj razlik. Robotske kosilnice so običajno bistveno večje od robotskih sesalnikov, kar posledično pomeni, da potrebujejo tudi večje akumulatorje ter močnejše pogonske motorje. Če primerjamo, robotski sesalnik iRobot Roomba je težak približno 3,6 kg, medtem ko je robotska kosilnica Husquvarna Automower težka približno 12 kg. Prav tako je zelo pomembno, da robot ne povози nepokošene trave s svojimi pogonskimi kolesi, saj bi bila tako posledično kvaliteta košnje bistveno slabša. Zato imajo robotske kosilnice običajno svoja pogonska kolesa nameščena na zadnji strani oziroma za vrtečim se rezilom, ki pokosi travo. Na prednjem delu imajo pa običajno zaradi stabilnosti nameščeno eno manjše vrteče se kolo. V primerjavi z robotskimi sesalniki, ki so običajno popolnoma okrogle oblike, so robotske kosilnice bolj podolgovate. Slika 2.14 prikazuje primer zgradbe tipične robotske kosilnice. Zaradi lokacije svojih pogonskih koles na zadnji strani ter podolgovate oblike to pomeni, da robotske kosilnice ne morejo narediti 360 stopinjskega obrata na mestu, brez da bi se zaleteli v kakšno oviro in obtičale. To pa pomeni, da potrebujejo robotske kosilnice nekoliko drugačno logiko premikanja pri odzivanju na ovire kot robotski sesalniki, ki zaradi svoje okrogle oblike in sredinsko nameščenih koles lahko naredijo 360 stopinjski obrat brez nevarnosti, da bi se zaleteli v kakšno oviro.



Slika 2.14: Primer zgradbe tipične robotske kosilnice [18].

## 2.2.2 Husqvarna Automower

Podjetje Husqvarna je bilo ustanovljeno že daljnega leta 1689, ko so izdelovali orožje za Švedsko vojsko. Od takrat je podjetje že precej zraslo ter razširilo svojo proizvodnjo in prodajo na šivalne stroje, kuhinjske aparate, motorna kolesa, kosilnice, motorne žage itd. Predvsem pa je podjetje znano po zelo dobrih kosilnicah, kar jim je delno omogočalo tudi dobra prodaja njihovih linij robotskih kosilnic. Podjetje je svojo prvo linijo robotskih kosilnic, imenovano Automower, ponudilo na globalni trg leta 1995. Vendar pa je podjetje doživelo razcvet prodaje svoje linije robotskih kosilnic šele leta 2004, ko so predstavili drugo generacijo linije Automower. Na dan pisanja je bilo podjetje Husqvarna vodilno na trgu v prodaji robotskih kosilnic, s prodajo, veliko približno 70.000 enot samo znotraj evropske regije [19]. Slika 2.15 predstavlja primer enega izmed njihovih robotov linije Automower.



Slika 2.15: Robotska kosilnica Husqvarna Automower 265 ACX [20].

Podjetje je pri drugi generaciji svoje linije robotskih kosilnic AutoMower naredilo kar precej izboljšav, ki pred tem še niso bile opažene pri nobeni drugi robotski kosilnici v tistem času. Ena izmed najpomembnejših izboljšav je, da je podjetje implementiralo več različnih načinov vračanja robota nazaj na svojo bazno postajo, kar odpravi problem puščanja opaznih sledi v travi, če se robot vedno vrača po eni in isti poti. In sicer robot AutoMower lahko svojo bazno postajo najde na enega izmed treh načinov [19]:

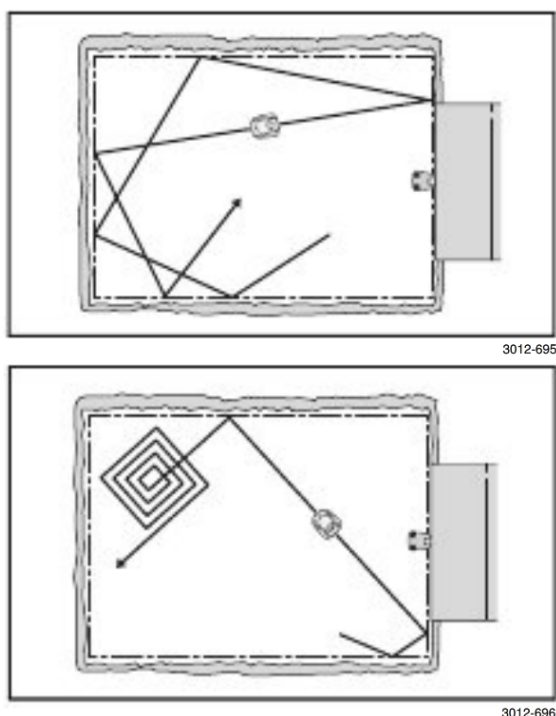
1. Najbolj osnovni način iskanja bazne postaje je, da se robot naključno premika toliko časa, dokler ne zazna prve lokacije robne zakopane žice oziroma prvega roba svojega območja delovanja. Ker je ta zakopana žica povezana v zanko na bazno postajo, robot samo enostavno sledi tej zakopani žici toliko časa, dokler ne pride nazaj do svoje bazne postaje, kjer se lahko ponovno napolni.
2. Drugi način je iskanje s pomočjo posebne vodilne zakopane žice, ki jo po navadi zakopljemo na sredino območja delovanja in vodi po najkrajši možni poti nazaj na bazno postajo. Pri tem načinu ima robot bistveno krajšo oziroma hitrejšo pot nazaj do svoje bazne postaje, kot pa če bi moral voziti po celotnem obrobju svojega območja delovanja, ki je lahko precej velik.

3. Tretji način je pa iskanje s pomočjo radijski frekvenc, in sicer oddaja bazna postaja poseben radijski signal, ki ga potem robot sprejema preko svojega sprejemnika in zna izračunati jakost tega signala. Na podlagi jakosti signala se poskuša robot premikati v tisto smer, v kateri jakost signala narašča toliko časa dokler ne pride nazaj na svojo bazno postajo. Torej se robot v bistvu relativno naključno premika toliko časa, dokler ne zazna signal bazne postaje. Glavna slabost tega načina iskanja je, da se lahko relativno dolgo naključno premika, preden bo končno zaznal signal bazne postaje. Najdaljša oddaljenost od bazne postaje, na kateri lahko robot še zazna signal bazne postaje, je približno 6–7 metrov.

Končni uporabnik robota lahko preko uporabniškega vmesnika določi, katerega izmed omejenih načinov iskanja bazne postaje naj se robot poslužuje. Uporabniški vmesnik omogoča tudi poljubno kombiniranje med seboj vseh treh načinov delovanja, s tem robot omogoča, da lahko končni uporabnik definira najbolj optimalen način vračanja nazaj na bazno postajo glede na obliko definirane območja njegovega vrta. Torej sam robot ne določa avtomatsko, kateri način naj bo uporabljen, ampak mora to definirati končni uporabnik [19].

Glede samega algoritma premikanja oziroma preiskovanja prostora robot deluje tako, da na začetku najprej sledi zakopani žici po robovih definirane območja, toliko časa dokler ne naredi celotnega obhoda nazaj do bazne postaje. Robot naredi vsaj enega ali več obhodov različne oddaljenosti od zakopane žice, točno število obhodov lahko določimo na samem robotu preko uporabniškega vmesnika. Po začetnem obhodu robot preklopi v naključni način premikanja oziroma se poskuša premikati v ravni liniji toliko časa, dokler ne naleti na oviro ali pa zazna rob definirane območja, kjer se ustavi, premakne nazaj za določeno dolžino ter se nato obrne za specifičen kot (oboje je nastavljivo preko uporabniškega vmesnika) in poskuša nadaljevati v ravni liniji. Po določenem času oziroma ko zazna, da mu zmanjkuje električne energije, pa preklopi v način iskanja bazne postaje (izklopi pogon rezil za

travo) ter se poskuša vrniti nazaj na bazno postajo na podlagi enega izmed prej opisanih načinov. Če robot zazna, da je prišel na nek določen odsek, kjer je procentualno trava precej višja kot drugje, preklopi v način premikanja po vzorcu kvadrata, zato da ne ostanejo šopi trave, ki so vizualno precej višji od ostale okolice oziroma temu odseku posveti več časa. Slika 2.16 grafično prikazuje opisan postopek premikanja. Ena izmed bistvenih razlik v primerjavi z robotskimi sesalniki je tudi ta, da robotska kosilnica Automower omogoča bistveno večjo stopnjo prilagajanja algoritma premikanja preko uporabniškega vmesnika kot robotski sesalniki, kjer običajno končni uporabnik sploh nima nobene opcije prilagajanja algoritma svojim potrebam [22]. Algoritem naključnega premikanja predpostavlja, da bo robot po določenem časovnem obdobju uspel preiskati celotno območje oziroma vsaj večino.



Slika 2.16: Algoritem premikanja robota Husqvarna Automower 265 ACX [22].

Glede napajanja robota imajo različni modeli različno veliko baterijo. Cenejši modeli (npr. Automower 210C) so namenjeni manjšim površinam in zadostujejo za približno 3–4 ure delovanja, medtem ko imajo dražji modeli večje baterije in so primerni za večje površine oziroma lahko delujejo dlje časa. Kot primer, model Automower 230ACX je sposoben z enim polnjenjem baterije pokositi površino, velikosti približno  $3/4$  ara. Eden izmed njihovih najnovejših modelov (Automower Solar Hybrid) ima na zgornji strani celo pritrjene sončne celice, ki omogočajo robotu polniti baterijo s pomočjo sončne energije med samim delovanjem. Ta tehnologija sončnih celic lahko bistveno podaljša avtonomijo baterije, vendar pa je pogojena z lepim vremenom [21].

## 2.3 Rezultat primerjave robotskih sesalnikov in kosilnic

Če primerjamo robotske kosilnice z robotskimi sesalniki, lahko ugotovimo, da imajo robotske kosilnice bistveno manj senzorjev za zaznavo okolice kot robotski sesalniki. Kar v bistvu pomeni, da so robotske kosilnice manj inteligentne kot robotski sesalniki in potrebujejo bistveno več predpriprav oziroma prilagoditev okolja, v katerem bodo delovale. Ker je zunanje okolje v splošnem precej kompleksnejše in raznoliko kot notranji prostori, oziroma imajo končni uporabniki lahko precej različna območja, ki jih želijo pokositi, tako potrebujejo robotske kosilnice tudi veliko več prilagajanja samih nastavitvev robota s strani končnega uporabnika zato, da lahko robot čim bolje deluje v zelenem okolju.

Torej na podlagi pregleda področja komercialno dostopnih široko proizvodnih robotov smo ugotovili, da je izdelava inteligentnega robota, ki bi dobro deloval v nekem splošnem zunanjem okolju bistveno težja kot za notranje okolje.



## Poglavje 3

# Komponente robota

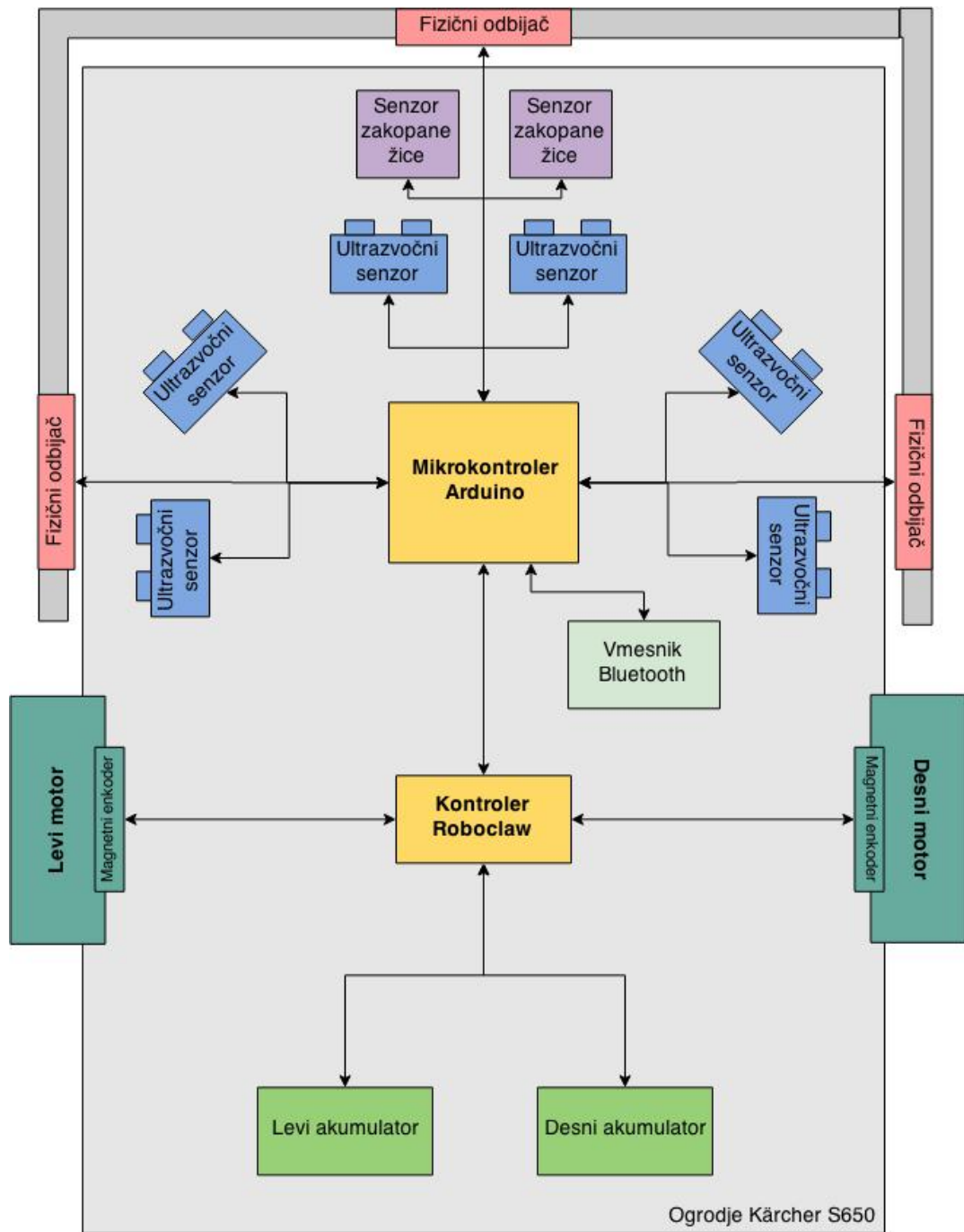
V tem poglavju bomo nekoliko bolj podrobno predstavili vse uporabljene komponente robota. Kot osnovo smo vzeli ročni pometlač podjetja Kärcher, ki smo ga nato opremili z motornim pogonskim sistemom ter dodali različne senzorje, na podlagi katerih smo nato razvili krmilno oziroma navigacijsko logiko. Za glavno krmilno vezje smo uporabili mikrokontroler Arduino. Pri celotnem postopku nameščanja senzorjev, nosilcev ter ostalih komponent smo se tudi močno poslužili tehnike 3D tiskanja, ki je zelo primerna za razvoj raznoraznih prototipov. Kateri del je bil narejen s pomočjo 3D tiskalnika, je podrobneje opisano v okviru posameznega poglavja. V našem primeru se je ta tehnologija 3D tiskanja izkazala za zelo uporabno ter nam omogočila izdelavo precej boljšega prototipa, kot pa če se je ne bi poslužili.

## 3.1 Oris sistema

Slika 3.1 predstavlja abstraktno shemo vseh uporabljenih komponent robota ter njihove medsebojne povezave. Naš robot torej vsebuje sledeče komponente:

1. Osnovno ogrodje Kärcher S650.
2. Mikrokontroler Arduino.
3. Kontroler motorjev Roboclaw.
4. Vmesnik bluetooth JY-MCU.
5. Šest ultrazvočnih senzorjev.
6. Dva senzorja zakopane žice.
7. Fizični odbijač.
8. Levi in desni motor z integriranim magnetnim enkoderjem.
9. Dva 12V 12Ah akumulatorja.

V nadaljevanju se nahaja nekoliko bolj podroben opis posameznih komponent ter njihova vloga v sistemu.



Slika 3.1: Abstraktna shema komponent robota, ter medsebojne povezave.

## 3.2 Osnovno ogrodje

Kot osnovno ogrodje smo vzeli že obstoječi ročni pometać podjetja Kärcher, in sicer bolj natančno model Kärcher S650. Kärcher je nemško podjetje, ki je bilo ustanovljeno že leta 1935 in je najbolj znano po svojih visokotlačnih čistilnikih, podjetje je celo razvilo prvi evropski visokotlačni čistilnik. Leta 1950 se je podjetje primarno osredotočilo na področje čistilnih strojev, med katere spadajo tudi ročni pometaći tal. Podjetje je v času pisanja te naloge predstavljalo eno izmed vodilnih podjetji na trgu čistilnih strojev, ki ima zelo kvalitetne ter visoko cenjene produkte [23].



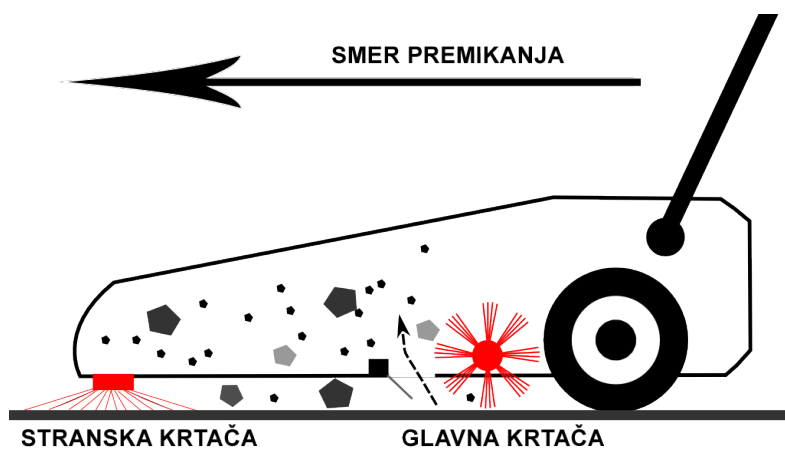
Slika 3.2: Kärcher S650 [24].

Ročni pometač Kärcher S650 (prikazan na sliki 3.2) v grobem deluje tako, da ima zabojnik ter krtače, ki so preko zobnika neposredno povezane na pogonska kolesa. Ko uporabnik preko ročajev premika pometač, se zaradi vrtenja koles vrtili tudi krtača, ki posledično nato pomete vse smeti v za to namenski zbiralnik. Ta model Kärcher S650 poleg glavne krtače, ki povzdigne smeti v zbiralnik, vsebuje tudi dve stranski krtači, ki sta namenjeni usmerjanju smeti iz robov stroja proti sredini oziroma proti glavni krtači. Te stranske krtače nam omogočajo tudi bolj natančno pometanje ob robovih sten saj so preko jermena povezane na glavno krtačo, kar jim omogoča pogon, oziroma da se vrtijo sinhrono z glavno krtačo. Model Kärcher S650 vsebuje 16 litrov velik zbiralnik za smeti in je težek približno 11kg [24]. Slika 3.3 prikazuje primer uporabe ročnega pometača, medtem ko slika 3.4 ponazarja opisan postopek pobiranja smeti. Slika 3.5 predstavlja ponazoritev opisane zgradbe ročnega pometača Kärcher S650.

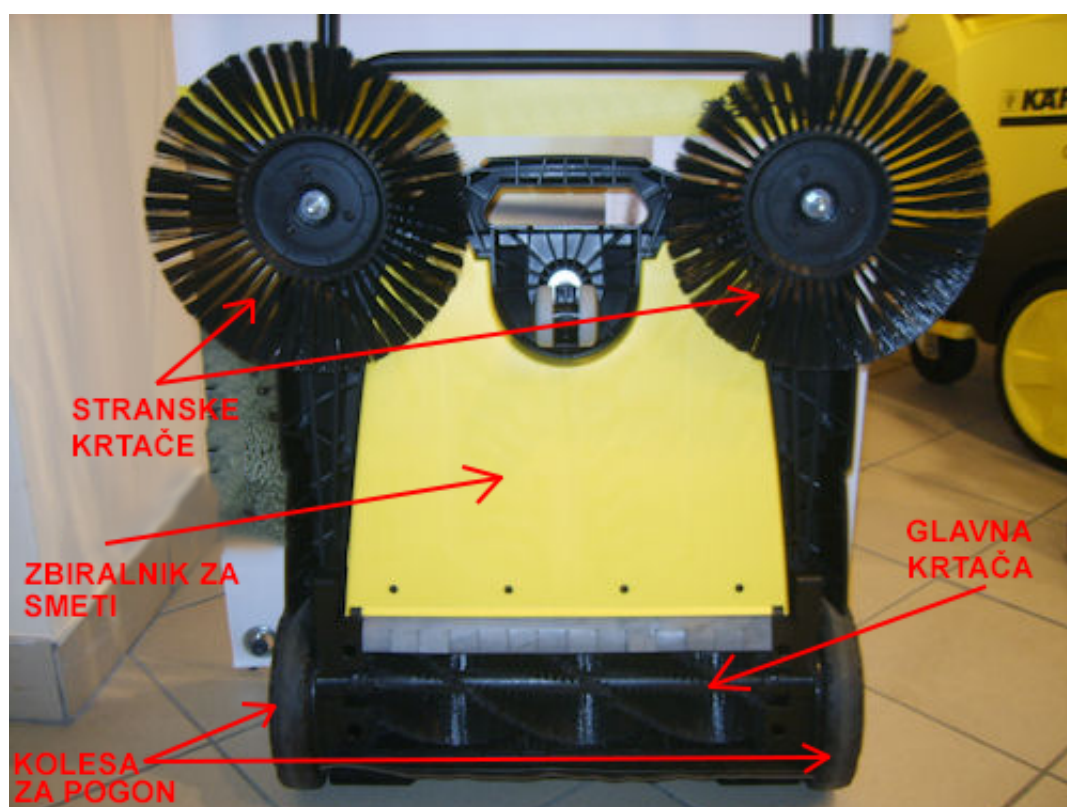
Praznjenje zbiralnika smeti je precej enostaven postopek, saj enostavno dvignemo zabojnik ter ga izpraznimo. Slaba lastnost je le, da nima nobenega indikatorja, kdaj je zbiralnik poln, tako da moramo periodično ročno preverjati polnost zbiralnika ter ga izprazniti, ko postane preveč poln.



Slika 3.3: Primer uporabe ročnega pometača Kärcher S650 [24].



Slika 3.4: Prikaz delovanja ročnega pometača Kärcher S650 [25].



Slika 3.5: Prikaz zgradbe pometača Kärcher S650.

### 3.3 Pogonski elektromotorji in krmilno vezje

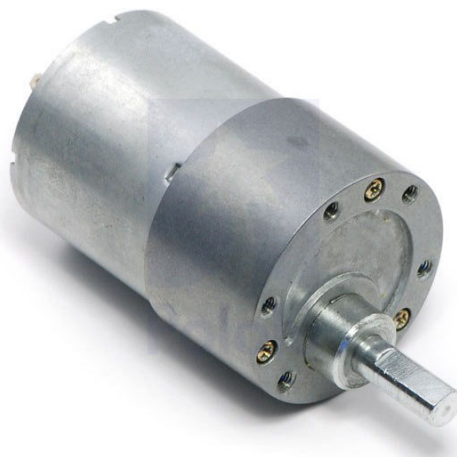
Prvi korak izgradnje mobilnega robota je zagotoviti primerno pogonsko tehniko oziroma motorje za premikanje. V našem primeru smo najprej razdrli obstoječe ogrodje pometača Kärcher S650 ter poskušali najti primerno mesto, kamor bi lahko namestili motorje. Obstoječi sistem pometača Kärcher S650 deluje tako, da ima na vsakem kolesu nameščen po en veliki zobnik, preko katerih je nato povezana glavna krtača, kar pomeni, da se ob vrtenju koles vrti tudi glavna krtača. Stranske krtače so pa preko posebnih jermenov povezane na kolesa sistema, kar tudi omogoča njihovo sinhrono vrtenje z vrtenjem kolesa sistema. Naš načrt je bil, da namestimo oziroma pritrdimo motorje na ohišje ter dodamo še en dodatni zobnik, ki je neposredno povezan na veliki zobnik na kolesu, kar potem omogoča tako vrtenje samega kolesa kot tudi glavne krtače ter stranskih krtač. To smo storili na obeh zadnjih kolesih. Slika 3.7 prikazuje pritrditev motorja na ohišje pometača ter notranji del kolesa. Dodatni zobnik, ki je nameščen na motorju in skrbi za sam pogon kolesa, smo dali izdelati iz železa pri namenskem ponudniku izdelave zobnikov v Velenju.

Za pogon robota smo uporabili dva motorja, nameščena vsakega na svoje zadnje kolo. Ker je prednje sredinsko kolo vrtljivo, nam to torej omogoča tako obračanje na mestu za poljubni kot, kot tudi premikanje v ravni liniji naprej in nazaj. Za znamko motorjev smo se odločili za podjetje Pololu, in sicer bolj natančno njihove 12 V krtačne motorje z že vgrajenim reduktorjem ter magnetnim enkoderjem<sup>1</sup>. Sam motor se pri 12 V vrti s hitrostjo 11000 vrtljajev na minuto, ki nato poganja reduktor s prestavnim razmerjem 30 : 1, kar pomeni, da je izhodna hitrost na osi reduktorja pri 12 V le še 350 obratov na minuto, vendar pa bistveno pridobimo pri navoru motorja. Končni maksimalni navor motorja z reduktorjem je tako 0,78 Nm, pri čemer ob tej maksimalni obremenitvi potrebuje motor kar 5 A toka. Ob minimalni obremenitvi oziroma prostem teku pa motor porabi le 350 mA toka [26].

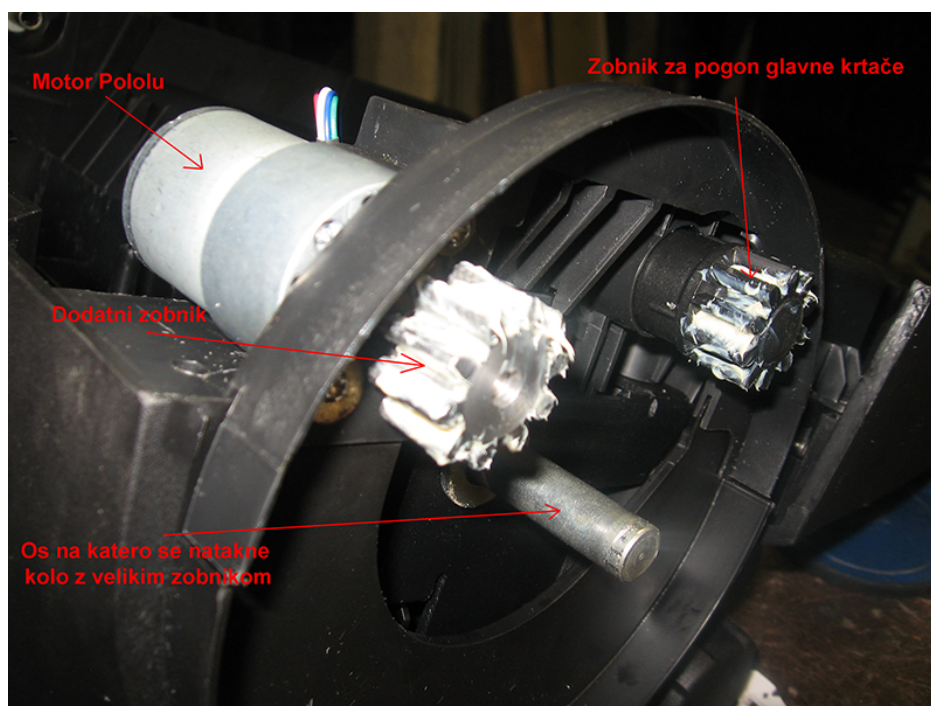
---

<sup>1</sup>Motor Pololu: <http://www.pololu.com/product/1443>





Slika 3.6: Motor podjetja Pololu z integriranim reduktorjem[26].



Slika 3.7: Namestitev motorja Pololu na ohišje ter notranji del kolesa.

Kot smo že omenili, smo motor pritrdili na ohišje robota ter nato na izhodno os iz reduktorja namestili še en dodatni namensko izdelan zobnik, ki se potem neposredno prilega na veliki zobnik kolesa. Ta manjši zobnik ima 10 zob, večji pa 40, kar pomeni, da imamo sedaj še dodatno prestavno razmerje 4 : 1 in se tako maksimalna hitrost vrtenja velikega zobnika na kolesu oziroma maksimalna hitrost vrtenja samega kolesa še dodatno zmanjša na 87 vrtljajev na minuto. Ob upoštevanju tega še dodatnega prestavnega razmerja ter navedenega navora reduktorja motorja lahko sedaj po formuli (3.1) izračunamo končni maksimalni navor pogonskega sistema robota, ki znaša 3,12 newton metra. Izračun:

$$\frac{\text{izhodni navor}}{\text{vhodni navor}} = \text{prestavno razmerje} = \frac{N_1}{N_2} \quad (3.1)$$

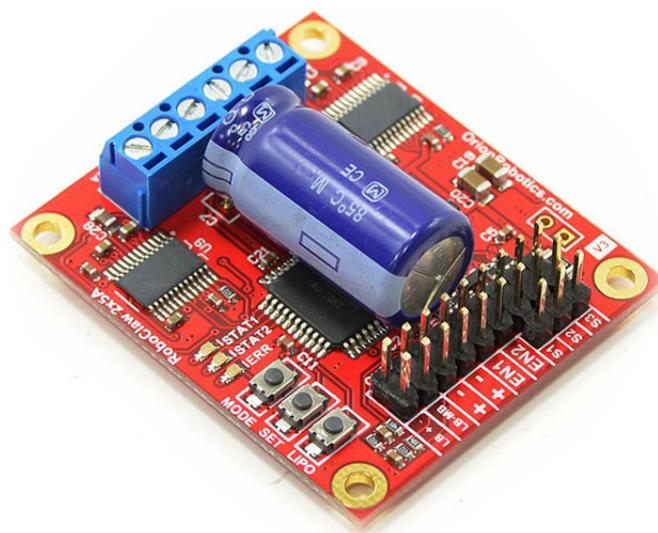
$$\text{izhodni navor} = \text{vhodni navor} * \frac{N_1}{N_2} = \frac{0,78Nm * 4}{1} = 3,12Nm \quad (3.2)$$

Za izračun maksimalne hitrosti pa potrebujemo najprej obseg kolesa, ki smo ga izmerili s pomočjo merilnega traka, in sicer znaša obseg enega kolesa 581 milimetra. Če to sedaj pomnožimo z izračunanim številom vrtljajev na minuto ter pretvorimo v ustrezne enote, dobimo, da je maksimalna hitrost našega robota 3,03 km/h, kar je v bistvu primerljivo s hitrostjo hoje odraslega človeka in več kot dovolj hitro za naše potrebe pometanja dvorišča oziroma za to namenskega robota. Sam ročni pometalč Kärcher S650 je bil prav tako zasnovan, da najbolje deluje pri povprečni hitrosti hoje odraslega človeka.

Ker je maksimalna tokovna obremenitev motorja 5 A, potrebujemo tudi ustrezno krmilno vezje, ki bo sposobno prenesti take obremenitve oziroma ne bo pregorelo, ko bomo dosegli maksimalno obremenitev. Podjetje Pololu poleg motorjev izdeluje tudi namensko vezja za krmiljenje teh motorjev, tako da smo se odločili za njihov produkt RoboClaw, ki je sposoben prenesti tokovno obremenitev do 5 A<sup>2</sup>. Namensko vezje RoboClaw je sposobno krmiliti dva motorja, oziroma je dvokanalno, pri čemer je maksimalna obremenitev 5 A na posamezni kanal. Poleg krmiljenja hitrosti ter smeri vrtenja motorjev ponuja RoboClaw vezje tudi priklop odometričnih senzorjev za zaznavanje

<sup>2</sup>Krmilno vezje motorjev RoboClaw: <http://www.pololu.com/product/1492>

dejanske hitrosti vrtenja motorja, pri čemer zna vezje tudi šteti impulze, tako da na izhodu lahko dobimo že dejansko izračunano hitrost vrtenja (pogoj je ustrezna konfiguracija vezja preko serijskega vmesnika). Vezje RoboClaw ponuja tudi regenerativno zaviranje ter detekcijo stanja akumulatorja. Ko pade napetost akumulatorja pod nek vnaprej določen nivo, vezje samodejno izklopi napajanje motorjev in s tem zaščiti akumulator pred prekomernim praznjenjem. Glede komunikacije vezja z glavnim krmilnim vezjem oziroma z mikrokrmilnikom Arduino pa vezje RoboClaw ponuja več različnih načinov, pri čemer smo se odločili za serijsko paketni način komunikacije, ker omogoča največ funkcij oziroma je najlažji za implementacijo z glavnim krmilnim vezjem Arduino. Za povezavo z mikrokrmilnikom Arduino potrebujemo namreč samo dve signalni žici [27]. Slika 3.8 predstavlja ponazoritev opisanega namenskega vezja RoboClaw.

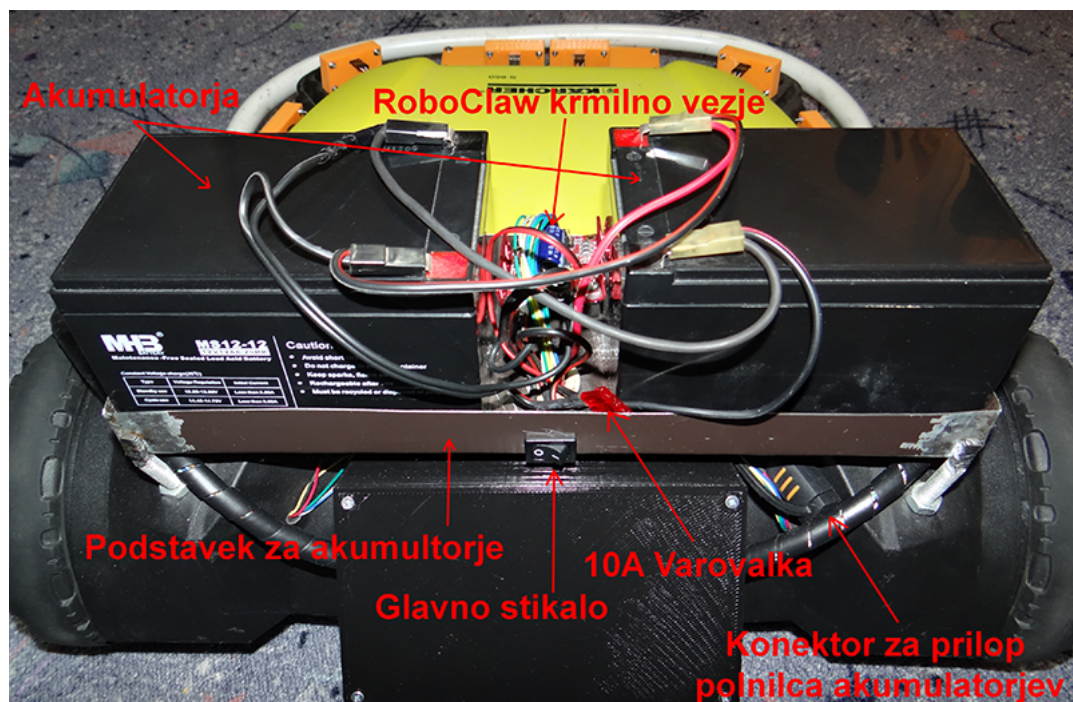


Slika 3.8: RoboClaw dvokanalno namensko krmilno vezje za krmiljenje motorjev [27].

## 3.4 Napajanje sistema

Za napajanje celotnega sistema smo se odločili za 12 V hermetično zaprte AGM akumulatorje, znamke MHB, ki premorejo kapaciteto 12 Ah. Ker smo želeli, da bo robot lahko neprestano deloval vsaj 2–3 ure, smo se odločili za dva vzporedno vezana omenjena akumulatorja. Vzporedna vezava akumulatorjev nam omogoča skupno kapaciteto sistema 24 Ah, kar bi morale zadostovati. Več kot dveh akumulatorjev si pa ne moremo privoščiti, saj en akumulator tehta 3,7 kg, kar predstavlja precejšnjo obremenitev za plastično ohišje. Glede same namestitve akumulatorjev na ohišje, smo se odločili, da je najbolj primerno mesto zadnji zgornji del robota, takoj nad zadnjo osjo, saj tako najmanj porušimo ravnovesje prvotnega ohišja. Drugi razlog je tudi v tem, da predstavlja zadnji del ohišja najmočnejši del ohišja in je tako najbolj sposobno prenesti težo akumulatorjev, ki skupaj znese kar 7,4 kg. Ker je zgornji del ohišja nekoliko ukrivljen oziroma ni popolnoma ploščat, smo tako morali iz pločevine narediti še en dodaten podstavek, ki smo ga potem neposredno namestili oziroma pričvrstili na zgornji del robota (s pomočjo navojnih palic). Na ta podstavek smo potem namestili oba akumulatorja ter krmilno vezje RoboClaw. Zaradi varnosti smo dodali še 10 A varovalko ter glavno stikalo za vklop oziroma izklop napajanja celotnega sistema. Slika 3.9 prikazuje opisano dejansko namestitev akumulatorjev na ohišje robota.

Za polnjenje akumulatorjev smo uporabili kar akumulatorski polnilnik, ki je bil namenjen polnjenju avtomobilskih akumulatorjev, ki imajo prav tako napetost 12 V, ter se ga da priklopiti na 220 V električno omrežje. Odločili smo se za akumulatorski polnilnik podjetja Ctek (model multi xs 3600), ker ima na voljo več različnih priključkov in smo lahko tako najbolj primeren priključek neposredno vkomponirali v robota. To nam v bistvu omogoča precej enostavnejši priklop na polnilec oziroma enostavnejše polnjenje robota. Slika 3.10 prikazuje ponazoritev omejenega polnilca ter priključkov.



Slika 3.9: Slika nameščenega celotnega napajalnega sistema robota.



Slika 3.10: Akumulatorski polnilec Ctek multi xs 3600.

## 3.5 Odometrija

Ena izmed najpomembnejši lastnosti skoraj vsakega robota je, da se zna premikati v ravni liniji naravnost naprej. Zaradi majhnih razlik v sami izdelavi motorjev ter različnih obremenitev motorjev ni vedno zagotovljeno, da se bosta oba motorja vrtela enako hitro, česar posledica je, da se robot ne bo vedno premikal v ravni liniji, temveč se bo čez določen čas začel premikati v loku. Ta problem odstopanja rešimo s pomočjo odometrije, in sicer tako, da namestimo na motorje posebne enkoderje, s pomočjo katerih lahko zaznamo hitrost vrtenja posameznega motorja, ter nato programsko kompenziramo to razliko v hitrosti vrtenja tako, da enemu motorju nekoliko dodamo ali pa odvezamemo nivo napajanja, da se bo začel vrteti enako hitro kot drugi motor. S tem zagotovimo, da se bo robot premikal v ravni liniji.

V grobem poznamo tri različne tipe enkoderjev:

1. Optični enkoderji – uporabljajo svetlobo ter disk z luknjami.
2. Mehanski enkoderji – uporabljajo disk, ki je električno prevoden ter posebne kontakte.
3. Magnetni enkoderji – uporabljajo magnetno polje za detekcijo vrtenja.

V našem primeru smo se odločili za magnetni enkoder, in sicer za takega, ki se pritrdi na zadnjo stran motorja. Zopet smo se odločili za podjetje Pololu, saj izdelujejo tudi magnetne enkoderje, ki se neposredno prilegajo na njihove motorje. Tako smo v bistvu v kompletu kupili motor, ki ima integriran tako reduktor kot tudi magnetni enkoder. Slika 3.11 predstavlja opisan komplet. Magnetni enkoder podjetja Pololu predstavlja dvokanalni magnetni enkoder, ki izkorišča Hallov efekt za detekcijo vrtenja. Deluje tako, da ima neposredno na osnovno os motorja na zadnji strani pritrjen magnetni disk ter posebno vezje, ki je sposobno zaznati prisotnost magnetnega polja na podlagi Hallovega efekta. Če upoštevamo oba kanala, je magnetni enkoder podjetja Pololu sposoben zaznati 64 pozicij na en polni obrat osnovne osi motorja. Slika 3.12 predstavlja ponazoritev signala, ki ga dobimo preko vezja.



Rumena barva predstavlja signal na prvem kanalu, modra pa na drugem. Če štejemo vse prehode na obeh kanalih, dobimo točno 64 pozicij na en obrat motorja. V našem primeru za to štetje impulzov skrbi vezje RoboClaw, ki šteje impulze na obeh kanalih obeh motorjev ter ustrezno in v realnem času pretvarja te vrednosti v hitrost vrtenja (obradi na minuto), ki jih razberemo iz glavnega krmilnega vezja ter na podlagi katere nato reguliramo nivo napetosti posameznega motorja, zato da lahko zagotovimo premikanje robota v ravni liniji [26].



Slika 3.11: Motor proizvajalca Pololu s integriranim reduktorjem in magnetnim enkoderjem [26].

Poleg reševanja problema premikanja robota v ravni liniji, bi lahko odometrijo uporabljali tudi za samo navigacijo robota oziroma bi lahko izračunali, koliko daleč se je robot premaknil od začetne pozicije. Vendar pa imamo pri tem problem natančnosti, saj se lahko na primer zgodi, da eno kolo nekoliko spodrsava na podlagi, kar v bistvu pomeni, da izračunana razdalja ni več 100 % natančna in če upoštevamo, da se ta napaka akumulira s časom, lahko hitro pridemo do zelo velikih odstopanj.



Slika 3.12: Primer signala magnetnega enkoderja podjetja Pololu [26].

### 3.6 Fizični odbijač

Fizični odbijač predstavlja enega izmed osnovnejših senzorjev robota, ki služi predvsem temu, da robot zazna, kdaj se je zaletel v kakšno oviro. Ta senzor v grobem deluje tako, da je premičen odbijač ponavadi pritrjen na prednji del robota ter ima dodatne vzvode, na koncu katerih so ponavadi posebna mikrostikala. V primeru potisne sile na odbijač se ta sila preko vzvodov prenese na mikrostikala, ki se potem ob zadostni sili sprožijo ter pošljejo električni signal na krmilno vezje. Tako v bistvu lahko zaznamo, kdaj se je robot zaletel v neko oviro.

V našem primeru smo za odbijač uporabili aluminijasto cev premera 25 mm, ki smo jo ukrivili v obliko prednjega dela ohišja ter nato pripravili ustrezne vzvode in naredili ustrezne luknje v ohišju. Odbijač smo na ohišje namestili s pomočjo pripravljenih vzvodov ter vzmeti tako, da se lahko giblje prosto navznoter v vse smeri. Odločili smo se za tri vzvode ter tri mikrostikala, enega spredaj ter še po enega na vsako bočno stran. Tako pokrijemo celotni prednji del, ne glede na to iz katere smeri pride sila. Ko pritismo na odbijač iz katerekoli smeri, se pomakne odbijač skupaj z vzvodi navznoter, ki



prenesejo to silo oziroma pritisnejo na mikrostikala. Ko je sila na mikrostikalih dovolj velika, steče preko mikrostikala električni tok, kar potem zaznamo na mikrokrmilniku, ki nam pove, da je bil sprožen eden izmed mikrostikal oziroma da se je odbijač pomaknil navznoter.

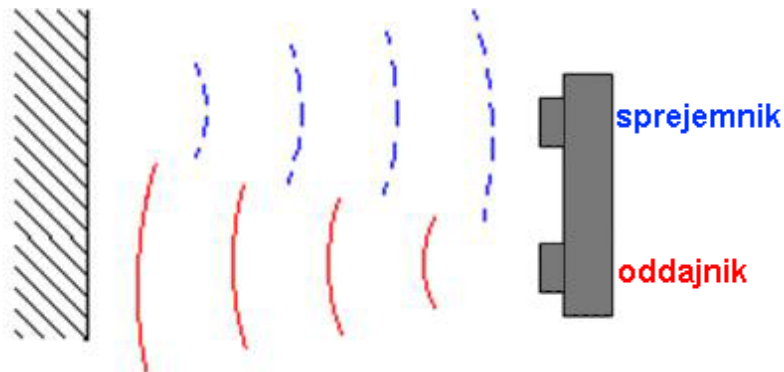
Glede samih mikrostikal je tudi zelo pomembna oddaljenost mikrostikala od konca vzvoda oziroma kakšno pot mora prepotovati vzvod, preden sproži mikrostikalo. S tem reguliramo, kolikšno silo na odbijač še dopuščamo, preden se mora robot ustaviti. Ta sila ne sme biti premajhna, saj ne želimo, da bi kakšne vibracije ali pa pospeševanje robota sprožilo mikrostikalo, prav tako pa tudi ne sme biti prevelika, ker drugače lahko pride do kakšnih poškodb odbijača oziroma robota na splošno. Torej smo za ta namen s pomočjo 3D tiskalnika narisali ter natisnili posebne vmesne elemente, ki smo jih pritrdili na ohišje ter nam omogočajo možnost poljubnega nastavljanja oddaljenosti mikrostikal od konca vzvodov. S tem smo pridobili možnost regulacije sile na odbijač, pri kateri se sprožijo mikrostikala. Na sliki 3.13 so ti nosilci obarvani z oranžno barvo. Slika 3.13 prav tako ponazarja končni rezultat namestitve fizičnega odbijača na ohišje.



Slika 3.13: Namestitev fizičnega odbijača na ohišje robota.

### 3.7 Ultrazvočni senzorji

Ultrazvočni senzorji se uporabljajo za detekcijo ovir in nam omogočajo, da zaznamo oviro še preden se robot dejansko zaleti vanjo. Delujejo na principu ultrazvoka, in sicer tako, da ima senzor ultrazvočni oddajnik ter ultrazvočni sprejemnik. Oddajnik oddaja ultrazvočni signal, sprejemnik pa posluša, kdaj se bo ta signal odbil od kakšne ovire ter pripotoval nazaj do sprejemnika. Na podlagi časa, ki ga je potreboval ultrazvočni signal, da se je odbil od ovire ter pripotoval nazaj do sprejemnika, ter na podlagi znane hitrosti potovanja zvoka ( $340 \text{ m/s}$ ), lahko izračunamo razdaljo ovire od senzorja. Slika 3.14 predstavlja ponazoritev opisanega pojava. Ultrazvočni senzorji se precej pogosto uporabljajo pri raznoraznih robotih, predvsem pa pri robotih za zunanjo rabo, saj niso občutljivi na sončno infrardečo svetlobo. Uporabljajo se celo pri večini parkirnih senzorjev na avtomobilih.



Slika 3.14: Delovanje ultrazvočnega senzorja [28].

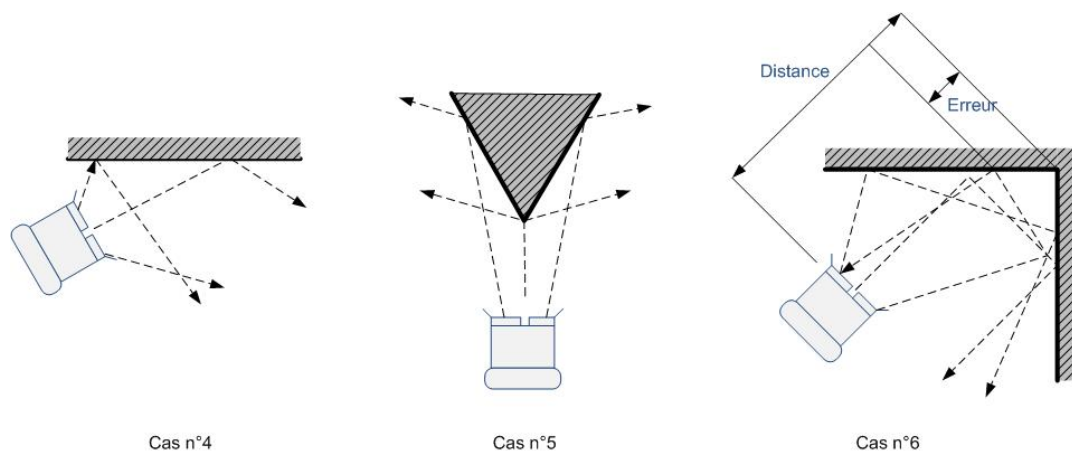
V grobem poznamo tri različne tipe senzorjev, ki nam omogočajo zaznavanje razdalje do ovir. In sicer so to ultrazvočni, infrardeči ter laserski senzorji. Tabela 3.1 predstavlja primerjavo med vsemi tremi tipi senzorjev ter njihove pomanjkljivosti oziroma prednosti.

	Ultrazvočni senzor	Infrardeči senzor	Laserski senzor
<b>Razdalja</b>	od 1 do 250 cm	od 5 do 80 cm	od nekaj milimetrov do več deset metrov, odvisno od modela
<b>Kotni razpon</b>	Stožec približno $30^\circ$	Stožec približno $5^\circ$	Skoraj povsem usmerjeno ( $0,5^\circ - 1^\circ$ )
<b>Natančnost</b>	Relativno natančno, natančnost pada z večanjem razdalje, kota, temperature zraka ter zračnega pritiska	Relativno natančno, natančnost pada z večanjem razdalje	Natančnost do nekaj centimetrov na razdalji večih metrov
<b>Cena</b>	poceni	poceni	precej dragi
<b>Občutljivost</b>	Občutljivost na zračni pritisk in temperaturo	Občutljivost na močne svetlobne vire, ki vsebujejo infrardečo svetlobo, kot je npr. sonce	Ne morejo zaznati objektov, ki odbijajo laserske žarke (npr. okna)

Tabela 3.1: Primerjava različnih senzorjev za detekcijo razdalje do predmetov [28].

Na podlagi primerjave med različnimi senzorji, ki so nam na voljo, smo se odločili za ultrazvočne senzorje predvsem zaradi njihove cenovne dostopnosti ter neobčutljivosti na sončno svetlobo. Večino časa bo namreč naš robot deloval zunaj na sončni svetlobi, kar pomeni, da so infrardeči senzorji zaradi občutljivosti slaba izbira, laserski senzorji so pa predragi za naš namen.

Ultrazvočni senzorji imajo tudi nekaj omejitev glede sposobnosti zaznavanja različnih tipov objektov oziroma ovir. Prva omejitev je dejanska gostota materiala, iz katerega je narejena ovira, namreč če je ovira premalo gosta gre ultrazvočni signal enostavno skozi oviro ter se ne odbije nazaj v sprejemnik (primer premalo goste ovire je npr. tkanina). Druga omejitev je oblika ovire, saj se zvok vedno odbije pri kotu, ki je enak vpadnemu, in če je ovira nepravilne oblike, se lahko zvok odbije ravno v napačno smer oziroma ne pride nazaj do sprejemnika. Slika 3.15 prikazuje nekaj primerov oblik ovir, pri katerih se zvok nepravilno odbije in lahko dobimo nepravilne rezultate za razdaljo.



Slika 3.15: Primer nepravilnega odboja zvoka pri uporabi ultrazvočnih senzorjev [28].

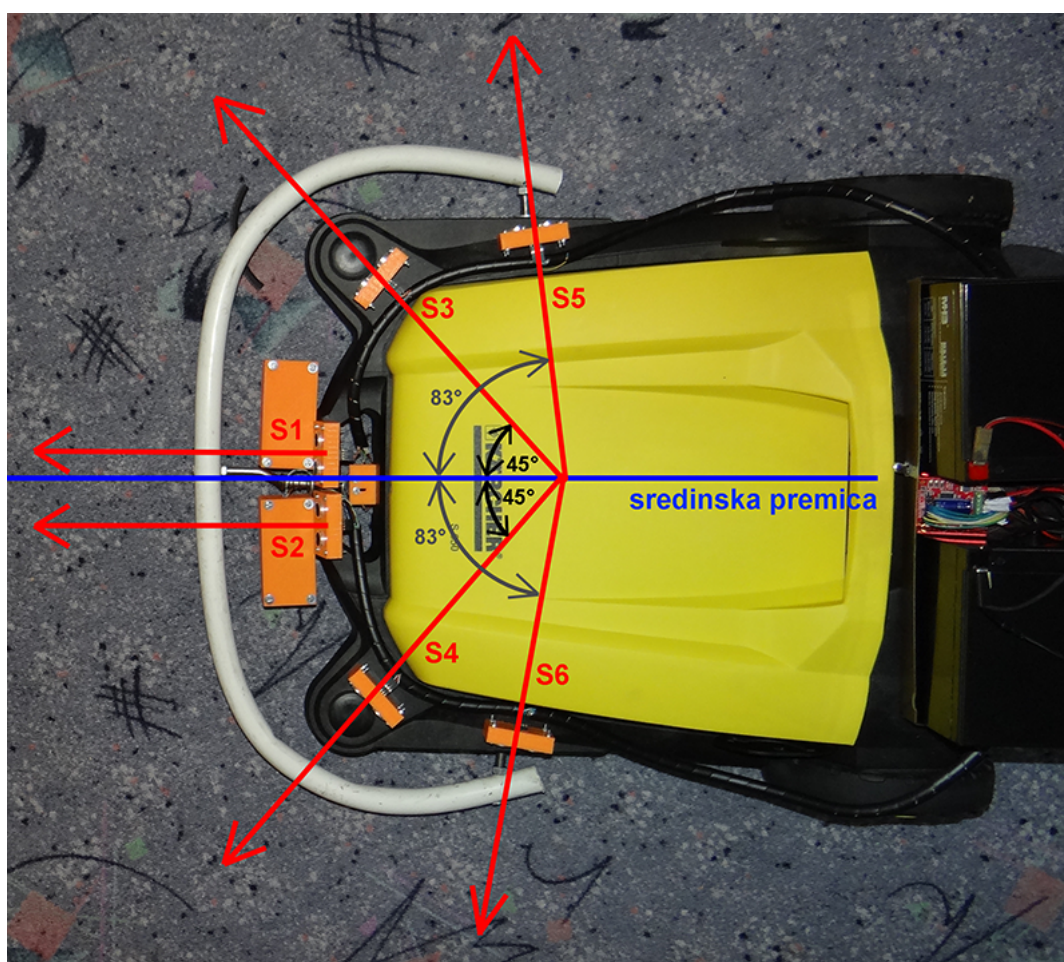
V našem primeru smo uporabili ultrazvočne senzorje HC-SR04, ki imajo neposredno na vezju vgrajen tako oddajnik kot sprejemnik ter jih je relativno enostavno priklopiti na mikrokrmilnik. Poleg napajanja potrebujemo samo še dve signalni žici, eno izhodno, s katero aktiviramo oddajnik, ter eno sprejemno, preko katere dobimo vhodni impulz, ko je sprejemnik zaznal odbiti signal. Slika 3.16 predstavlja ponazoritev opisanega ultrazvočnega senzorja HC-SR04.



Slika 3.16: Ultrazvočni senzor HC-SR04.

Naš cilj pri načrtovanju oziroma izgradnji robota je bil, da bi z ultrazvočnimi senzorji poskušali pokriti čim večji razpon na prednjem delu robota. Kajti, če dovolj kmalu zaznamo oviro, lahko pravočasno upočasnimo oziroma preprečimo, da bi se robot zaletel v oviro s polno hitrostjo, s čimer zmanjšamo možnost kakšnih mehanskih poškodb. Da bi pokrili čim večji razpon na prednjem delu, samo se odločili za kar 6 ultrazvočnih senzorjev HC-SR04 ter jih enakomerno porazdelili na sprednji del robota. In sicer smo jih razporedili tako, da smo namestili en par ultrazvočnih senzorjev, ki so usmerjeni naravnost naprej (S1 in S2), po en par, ki je umerjen pod kotom  $45^\circ$  (S3 in S4) ter po en par, ki je usmerjen pod kotom  $83^\circ$  (S5 in S6) glede na sredinsko premico. Slika 3.17 prikazuje opisano porazdelitev ter izbrano sredinsko premico. Kot smo že omenili, je vsak senzor sposoben pokrivati približno  $30^\circ$  razpon, ker pomeni, da kljub temu da imamo kar šest ultrazvočnih sen-

zorjev, še vedno ostajajo vmesne luknje v razponu, ki jih ne pokriva noben ultrazvočni senzor. Kar pomeni, da obstaja možnost, da v specifičnih primerih, če je ovira ravno pravšne oblike ter velikosti in se ji robot približuje pod ravno pravim kotom, je ne bomo zaznali z ultrazvočnimi senzorji. Vendar pa jo bomo zaznali s fizičnim odbijačem, tako da so v bistvu take luknje v razponu ultrazvočnih senzorjev tudi povsem sprejemljive oziroma ne bo zaradi njih odpovedal algoritem detekcije ovir. Slika 3.18 prikazuje opisan razpon pokrivanja ultrazvočnih senzorjev.



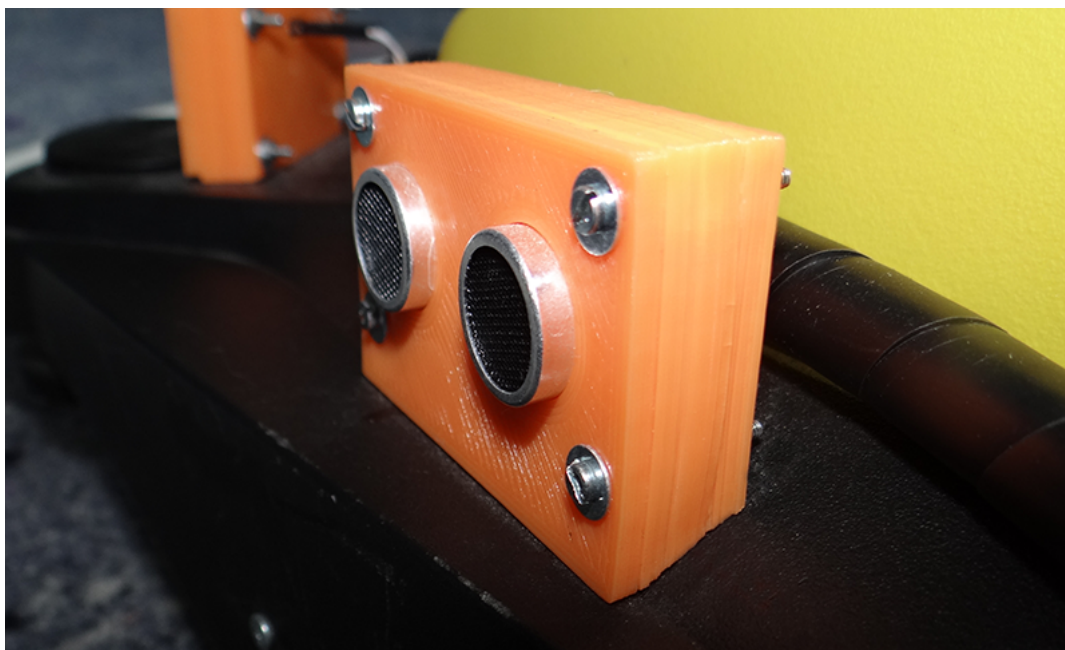
Slika 3.17: Usmerjenost ultrazvočnih senzorjev glede na sredinsko premico.





Slika 3.18: Območje oziroma razpon, znotraj katerega lahko s pomočjo ultrazvočnih senzorjev zaznamo oviro ter njeno oddaljenost (zelena barva).

Glede namestitve ultrazvočnih senzorjev na osnovo robota smo se zopet poslužili tehnike 3D tiskanja. Narisali ter natisnili smo ohišje, kjer se opisan senzor HC-SR04 čim bolj natančno prilega ter nam omogoča, da ga namestimo na osnovo robota pod ravno pravim kotom. Senzor mora biti usmerjen tako, da je čim bolj vzporedno poravnan s tlemi. Ker osnova robota ni povsod popolnoma ravna smo morali pripraviti ohišje senzorja tako, da ima spodnji del ustrezno prilagojen osnovi. Vsak par senzorjev smo namestili na drugi del osnove, ter morali tako pripraviti tri tipe ohišij z različnim spodnjim delom glede na podlago namestitve. Slika 3.19 prikazuje eno izmed opisanih ohišij, ki vsebuje ultrazvočni senzor HC-SR04.



Slika 3.19: Senzor HC-SR04 v namenskem ohišju, ki smo ga natisnili s pomočjo 3D tiskalnika.

### 3.8 Senzor zakopane žice

S pomočjo ultrazvočnih senzorjev in fizičnega odbijača smo tako pokrili večino primerov ovir, na katere lahko naš robot naleti na nekem običajnem dvorišču. Vendar ker zunanji prostor oziroma dvorišče ni nujno vedno neko omejeno področje (npr. do dvorišča vodi cesta ali pa je del dvorišča travnata površina) potrebujemo še nek dodatni senzor, s pomočjo katerega lahko omejimo robotu njegovo območje delovanja. Tako lahko na primer preprečimo robotu, da bi šel na cesto, da bi šel na travnato površino ali pa kakšno drugo površino, kjer ne želimo, da deluje. Torej za namen definiranja oziroma omejevanja območja delovanja robota smo se odločili za senzor zakopane žice.

Senzor zakopane žice deluje v grobem tako, da na robove območja, kjer želimo, da robot deluje, zakopljemo oziroma položimo posebno žico, ki jo priklopimo na napajalnik. Na robota pa namestimo posebne senzorje, ki za-



znajo prisotnost te zakopane žice. S tem zna robot v bistvu zaznati, kdaj je prišel do roba njegovega območja delovanja ter mora obrniti. Enake senzorje za zaznavanje robov območja uporabljajo tudi robotske kosilnice. Bolj podroben opis delovanja senzorja se nahaja v Poglavju 2.2.

Zelo podoben princip zakopane žice ter senzorja prisotnosti zakopane žice se uporablja tudi pri električnih ovratnicah za pse oziroma pri nevidnih električnih ograjah za pse. Pri teh ograjah v bistvu zakopljemo žico, ki oddaja omejeni radijski signal v zemljo, na psa pa namestimo posebno ovratnico, ki najprej začne piskati, ko se pes približa robu njegovega omejenega področja, ko pa pride preblizu oziroma prestopi to območje, pa ovratnica celo odda električne impulzne spodbude, kar načeloma prepreči psu prestop območja. To je v bistvu povsem enaka funkcionalnost, kot jo tudi potrebujemo pri našem robotu. Ker smo želeli privarčevati čim več časa pri izdelavi senzorja zakopane žice za našega robota, smo tako torej kar kupili sistem nevidne električne ograje za pse ter predelali te omenjene ovratnice v senzorje zakopane žice, ki smo jih potem namestili na našega robota.

Vzeli smo obstoječi komplet nevidne električne ograje za pse, prikazan na sliki 3.20, ki vsebuje žico (1), napajalnik (2) ter ovratnico (3). Žico ter napajalnik smo ohranili nespremenjen ter položili žico na robove našega dvorišča in jo priklopili na napajalnik. Ovratnico smo predelali v senzor ter ga namestili na našega robota.

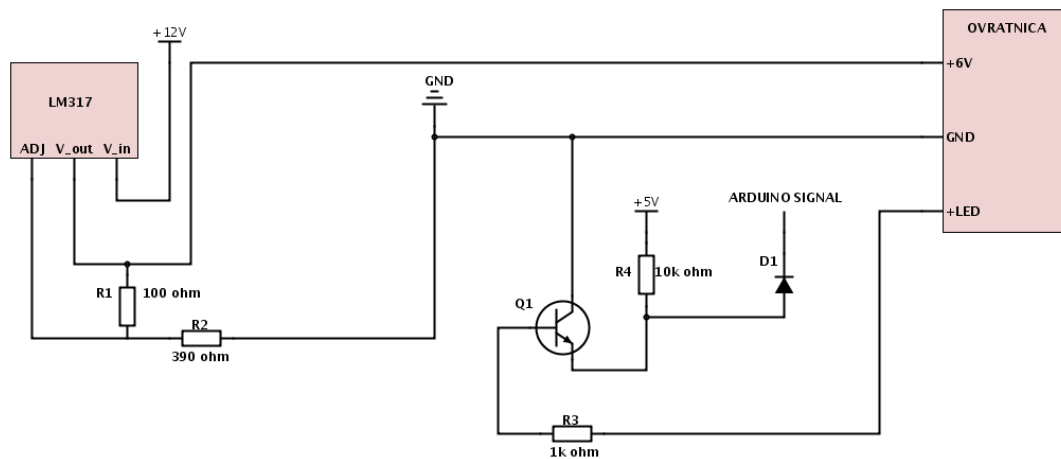
Torej smo najprej ovratnico razdrli ter odstranili vezje oziroma komponente iz plastičnega ohišja. Ugotovili smo, da ovratnica vsebuje:

1. LED diodo, ki začne utripati, ko se ovratnica približa zakopani žici.
2. Zvočnik, ki začne oddajati zvočni signal, ko se ovratnica približa zakopani žici.
3. Elektrošok generator, ki se sproži, ko je ovratnica preveč blizu zakopani žici.
4. 6V baterijo za napajanje celotnega sistema.

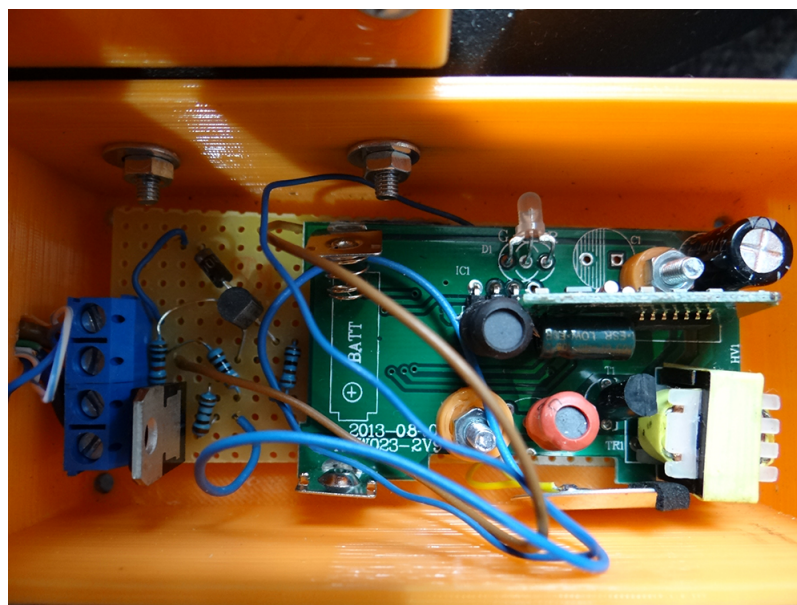


Slika 3.20: Komplet nevidne električne ograje za pse.

Na podlagi te ugotovitve smo se odločili, da bi bil najlažji način, da naredimo še eno dodatno vezje, ki je sposobno ojačati signal LED diode ter nato ta signal pripeljati na glavno krmilno vezje v obliki digitalnega vhodnega signala (logična enica oziroma logična ničla). Ta signal LED diode smo ojačali oziroma pretvorili v omejeno ustrezno obliko s pomočjo TIP120 močnostnega NPN tranzistorja ter nekaj uporov. To dodatno vezje vsebuje tudi še napetostni pretvornik, ki je sposoben pretvoriti 12 voltov vhodne napetosti iz akumulatorjev našega robota v 6 V napetost, ki jo potrebuje vezje ovratnice (prej je to napetost dobavljala baterija). Napetostni pretvornik je bil realiziran s pomočjo LM317 nastavljivega napetostnega regulatorja ter uporov ustreznih vrednosti tako, da na izhodu dobimo točno 6 voltov izhodne napetosti. Slika 3.21 predstavlja shemo omejenega dodatnega vezja, slika 3.22 pa predstavitev končne oblike dodatnega zgrajenega vezja.

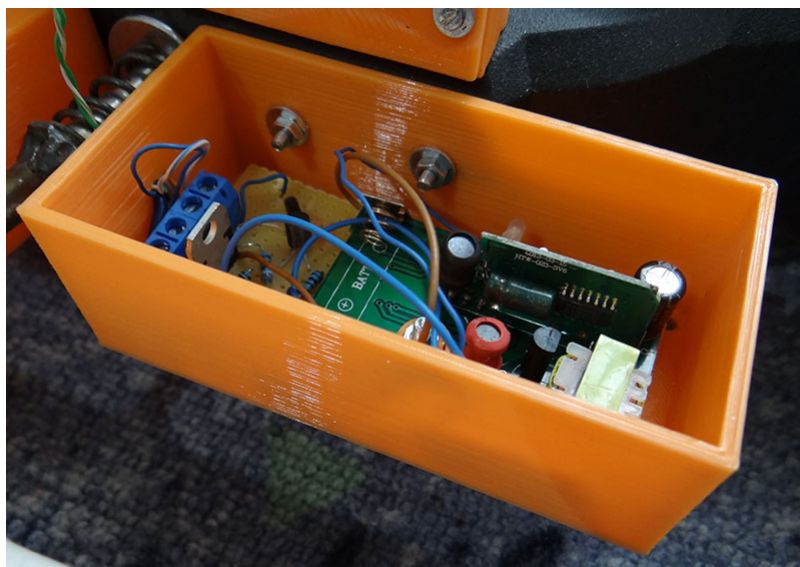


Slika 3.21: Shema dodatnega vezja za senzor zakopane žice.



Slika 3.22: Končna verzija vezja senzorja zakopane žice.

Ker imamo sedaj še eno dodatno vezje, to pomeni, da senzorja ne moremo več namestiti nazaj v originalno plastično ohišje in tako potrebujemo novo ohišje, v katerega lahko zapakiramo celotno vezje senzorja. Tukaj smo se ponovno poslužili tehnike 3D tiskanja ter narisali in natisnili posebno ohišje, kjer se izdelan senzor zakopane žice tesno prilega ter nam omogoča, da ga namestimo oziroma pritrdimo na osnovo robota. Odločili smo se, da bomo na sprednji del robota namestili kar dva senzorja zakopane žice. Za dva senzorja smo se odločili zaradi zmanjševanja možnosti napake oziroma, da povečamo verjetnost, da bo robot pravočasno zaznal zakopano žico ne glede na to iz katere smeri se ji približuje. Slika 3.23 prikazuje opisano natisnjeno ohišje senzorja, ki vsebuje zgrajeno dodatno vezje ter prvotno vezje ovratnice, slika 3.24 pa prikazuje oba nameščena senzorja na osnovo robota.



Slika 3.23: Senzor zakopane žice v ohišju, ki je bil natisnjen s pomočjo 3D tiskalnika.



Slika 3.24: Namestitev senzorjev zakopane žice na osnovo robota.

### 3.9 Komunikacijski sistem

Ker smo med samim razvojem rešitve želeli tudi analizirati oziroma nadzirati delovanje samega robota, smo na robota namestili še dodatni komunikacijski sistem, ki nam omogoča dvosmerno komunikacijo med osebnim računalnikom in robotom ali pa med pametnim telefonom in robotom. In sicer smo se odločili za dvosmerno komunikacijo na podlagi tehnologije bluetooth. Komunikacijski sistem je v bistvu sestavljen iz vmesnika bluetooth, ki se priklopi neposredno na glavno krmilno vezje in omogoča vzpostavitev serijske dvosmerne komunikacije s katerokoli napravo, ki podpira bluetooth serijsko povezavo. Mi smo se odločili za vmesnik JY-MCU, ki je precej cеноvno dostopen ter prav namensko zgrajen oziroma načrtovan za integracijo z našim glavnim krmilnim vezjem Arduino. Vmesnik samo enostavno priključimo na napajanje ter na serijski komunikacijski kanal (TXD in RXD signala) glavnega krmilnega vezja Arduino. Slika 3.25 predstavlja ponazoritev uporabljenega bluetooth vmesnika JY-MCU.



Slika 3.25: Bluetooth vmesnik JY-MCU.

Komunikacijski sistem smo primarno uporabljali za analizo delovanja samega robota, kar nam je precej olajšalo odkrivanje napak v izvorni kodi, saj smo lahko nadzorovali stanje robota oziroma stanje izvajanja algoritmov na daljavo med samim delovanjem robota zunaj na dvorišču, brez da bi za to morali robota ustaviti ter ga preko kabla priklopiti na računalnik. Poleg analize delovanja robota nam ta komunikacijski sistem omogoča tudi, da lahko na daljavo posodabljammo izvorno kodo robota z novejšimi verzijami ter pošiljamo robotu različne ukaze oziroma zgradimo uporabniški vmesnik.

### 3.10 Glavno krmilno vezje

Za glavno krmilno vezje smo se odločili za odprtokodni mikrokrmilnik podjetja Arduino. Platforma Arduino je bila prvič predstavljena leta 2005 in je bila zasnovana tako, da omogoča cenovno ugodno in enostavno izgradnjo različnih sistemov, ki zaznavajo in upravljajo z okoljem preko raznoraznih senzorjev in pogonskih sistemov. Platforma omogoča, da preko enostavnega integriranega razvojnega okolja na osebnih računalnikih pišemo programe v programskem

jeziku C ali C++ in potem ta program enostavno samo naložimo na mikrokontroler Arduino preko USB vmesnika. Glavni namen podjetja Arduino je pripraviti čim bolj enostavno in uporabniku prijazno platformo za razvoj različnih prototipov, kar vključuje tudi prototipne robote [29]. Poleg omejene enostavnosti in namembnosti za izdelavo prototipov, smo se za platformo Arduino odločili tudi zaradi zelo široke skupnosti uporabnikov in ker je za to platformo na voljo že ogromno različnih senzorjev, ki so bili načrtno zasnovani tako, da čim bolje delujejo z omejeno platformo, kar nam je bistveno olajšalo oziroma pohitrilo celoten proces izdelave robota.

Obstaja več različnih vrst platforme Arduino, ki se med seboj razlikujejo predvsem po velikosti in zmogljivosti. Tabela 3.2 predstavlja primerjavo nekaj teh modelov.

	Arduino Nano	Arduino UNO	Arduino Mega
<b>Mikrokontroler</b>	ATmega328	ATmega328	ATmega2560
<b>Število digitalnih V/I</b>	14	14	54
<b>Število analognih vhodov</b>	8	6	16
<b>Velikost pomnilnika</b>	32 KB	32 KB	256 KB
<b>Fizična velikost (š x d)</b>	1,8 cm x 4,3 cm	5,3 cm x 6,8 cm	5,3 cm x 10,1 cm

Tabela 3.2: Primerjava različnih modelov platforme Arduino [29].



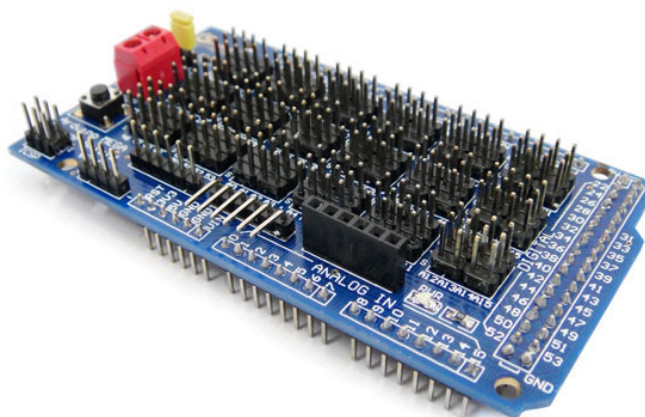
V našem primeru smo se odločili za model Arduino Mega, predvsem zaradi tega, ker smo potrebovali več kot 14 digitalnih vhodov/izhodov, ki jih ponujata Arduino Nano in UNO. Ker je naš robot relativno velik, nas sama fizična velikost Arduino ploščice ni omejevala in tako je bila edina logična odločitev model Arduino Mega. Slika 3.26 ponazarja uporabljeno ploščico Arduino Mega.



Slika 3.26: Ploščica Arduino Mega [29].

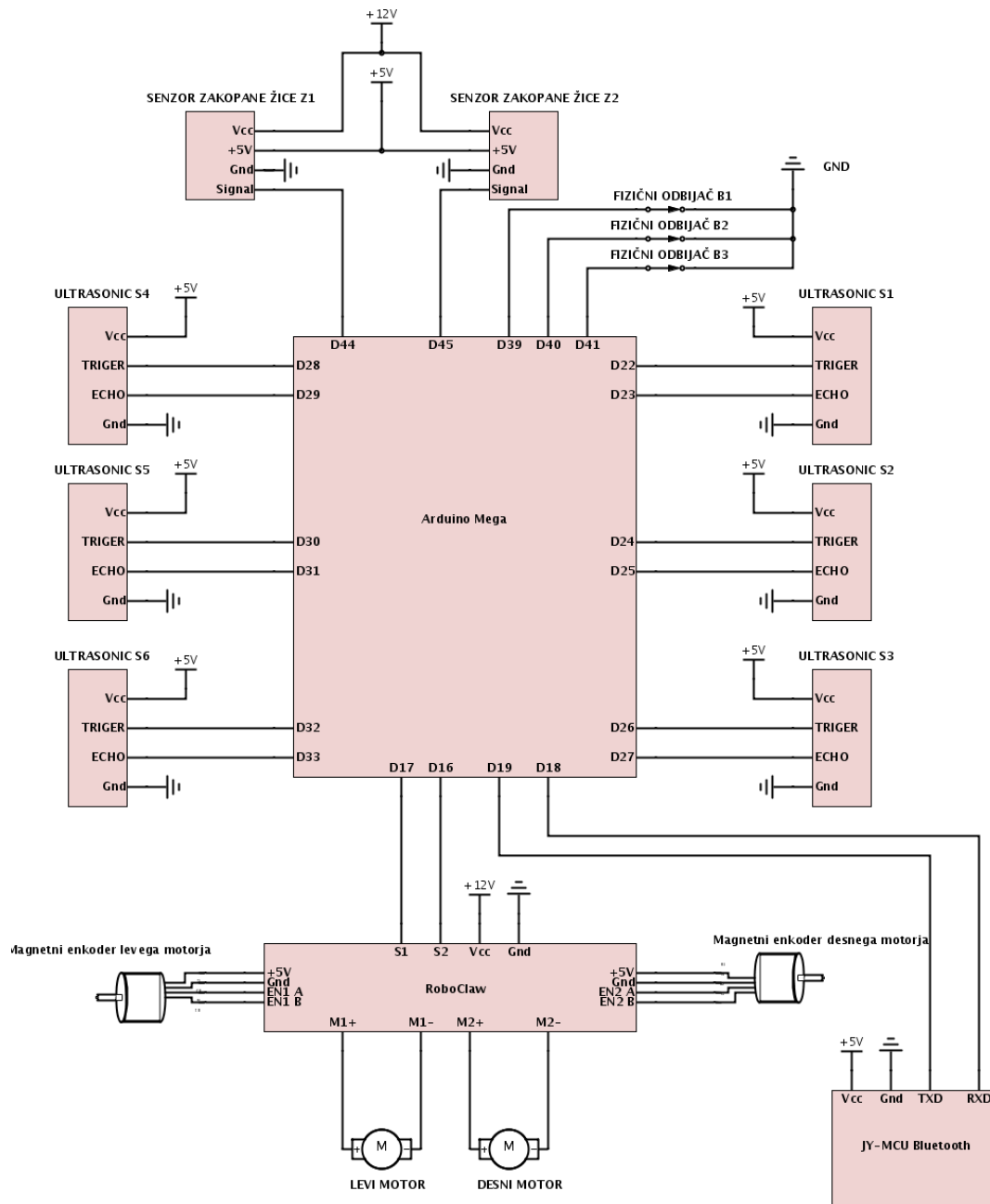
Na ploščico Arduino Mega smo ustrezno povezali vse v prejšnjih poglavjih opisane senzorje ter s pomočjo razvojnega okolja sprogramirali vso ustrezno krmilno logiko, ki jo bomo nekoliko podrobneje opisali v naslednjem poglavju. Zaradi lažjega priklapljanja senzorjev na Arduino Mega ploščico smo se poslužili še dodatne razširitve Arduino Mega Sensor Shield (slika 3.27), ki omogoča bolj enostavno in pregledno priklapljanje senzorjev ter ne vsebuje nobene dodatne logike. To razširitev smo namestili na vrh ploščice Arduino Mega ter nato preko nje povezali vse potrebne kable vseh prej opisanih senzorjev. Slika 3.28 prikazuje končno shemo celotnega vezja sistema.





Slika 3.27: Dodatna razširitev Arduino Mega Sensor Shield [29].

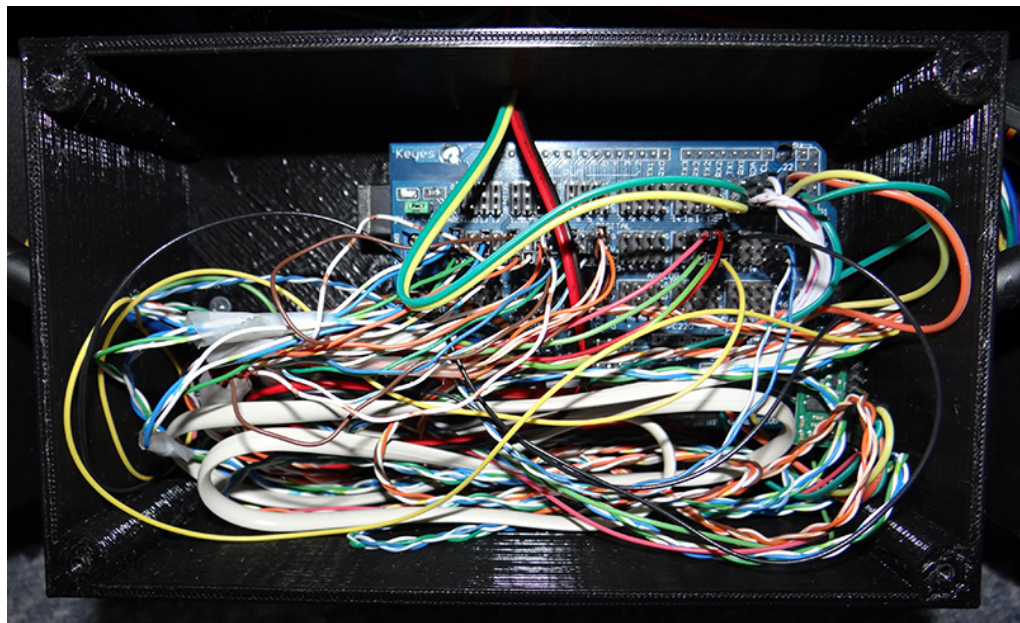
Zaradi relativno velikega števila senzorjev se je nabralo kar nekaj kablov, ki smo jih ustrezno organizirali ter lepo speljali po plastičnih ceveh na robovih osnove robota. Glede same lokacije namestitve glavnega krmilnega vezja smo se pa odločili, da je najbolj primerno mesto zadnja stran robota, takoj pod opisanim nosilcem za akumulatorje. Glavni razlog, da smo se odločili za to lokacijo, je predvsem v tem, da je bilo tam še največ prostora ter je tako tudi glavno krmilno vezje relativno blizu akumulatorjev in ne potrebujemo preveč dolgih napajalnih kablov. Zato da smo lahko glavno krmilno vezje namestili na samo osnovo robota smo se zopet poslužili tehnike 3D tiskanja, in sicer smo narisali in natisnili škatlico dimenzije  $17\text{cm} \times 10\text{cm} \times 7\text{cm}$  (*dolžina*  $\times$  *širina*  $\times$  *visina*), v katero se je ustrezno prilegalo glavno krmilno vezje z dodatno razširitvijo za senzorje, ter hkrati še prej opisan bluetooth adapter. To škatlico smo skupaj z vsemi komponentami namestili na zadnji del robota, kot to prikazuje slika 3.29. Slika 3.30 pa prikazuje nameščene komponente znotraj natisnjene škatlice oziroma ohišja.



Slika 3.28: Shema vezave celotnega sistema robota.



Slika 3.29: Namestitev glavnega krmilnega vezja na osnovo robota.



Slika 3.30: Škatlica oziroma ohišje glavnega krmilnega vezja z vsemi komponentami.



## Poglavje 4

# Krmilna logika robota

V tem poglavju bomo nekoliko bolj podrobno opisali razvito končno verzijo krmilne logike za našega robota. Kot smo že omenili, smo za “možgane” našega robota uporabili platformo oziroma mikrokrmilnik Arduino, ki ga lahko programiramo v programskem jeziku C/C++. Tako bomo torej najprej na kratko opisali uporabljeno razvojno okolje, s pomočjo katerega smo razvili celoten sistem ter kako smo si nekoliko olajšali postopek razvijanja ter testiranja na omejenem mikrokrmilniku. Nato bomo opisali končno verzijo razvitega sistema krmilne logike našega robota, razdeljeno na funkcionalne sklope oziroma podsisteme zaradi lažjega razumevanja.

## 4.1 Razvojno okolje

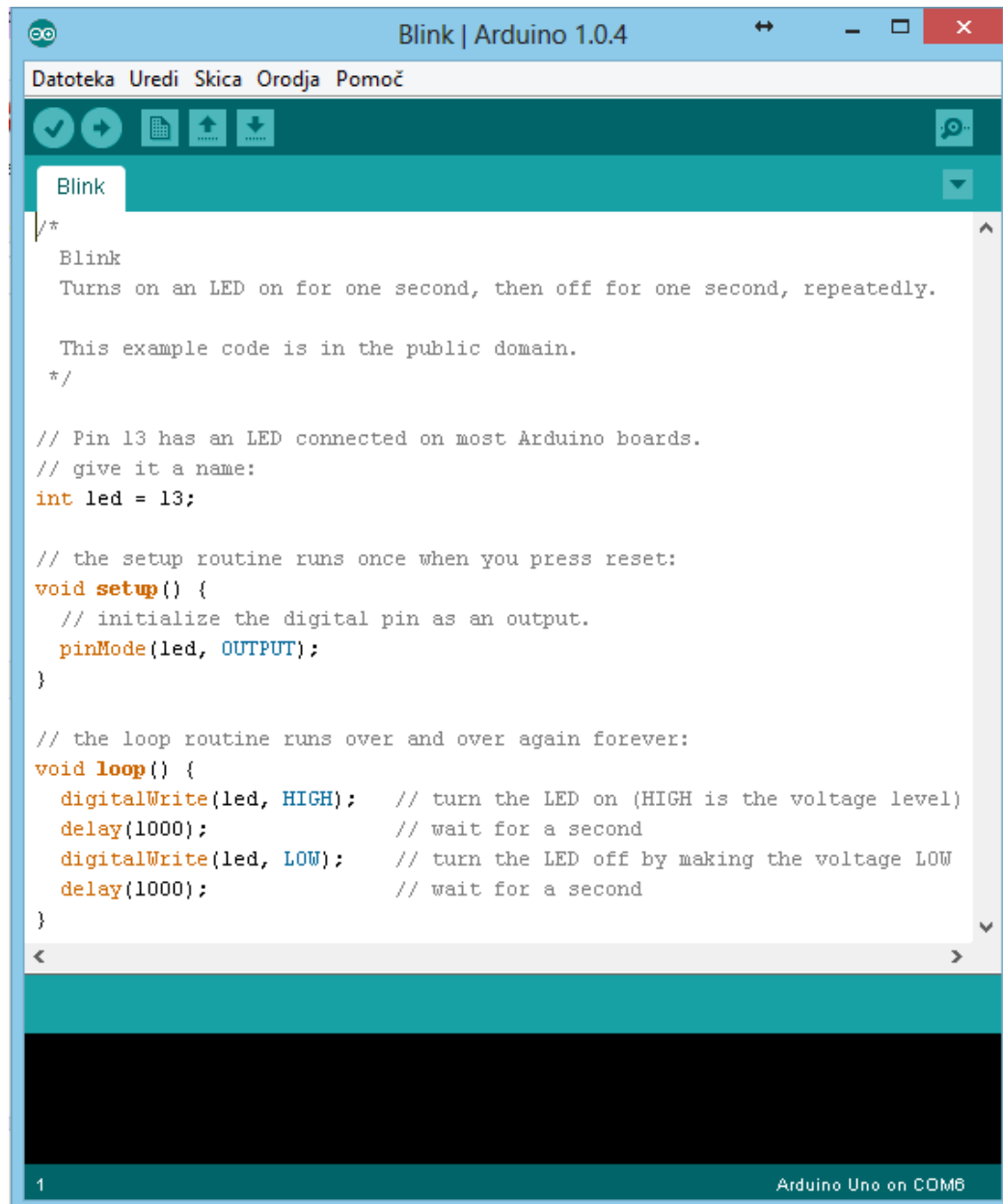
Podjetje Arduino ponuja svoje odprtokodno integrirano razvojno okolje<sup>1</sup>, poimenovano Arduino IDE, ki vsebuje:

1. Tekstovni urejevalnik, v katerem lahko pišemo C/C++ krmilno logiko.
2. Prevajalnik, ki je sposoben prevesti napisano C/C++ programsko kodo v strojno kodo, ki jo mikrokrmilnik Arduino razume.
3. Nalagalnik, ki je sposoben prevedeno strojno kodo naložiti v pomnilnik mikrokrmilnika Arduino preko povezave USB COM.
4. Gonilnike za platformo Arduino, zato da osebni računalnik prepozna platformo preko povezave USB.
5. Konzolo za serijsko komunikacijo z mikrokrmilnikom Arduino preko serijskega vmesnika USB COM.
6. Nekaj že napisanih različnih programskih knjižnic in primerov.

Arduino IDE orodje je bilo spisano v programskem jeziku Java, kar mu omogoča, da ga lahko poganjamo na različnih operacijskih sistemih (Windows, Linux in Mac OS X). Orodje je v bistvu izpeljanka podobnega razvojnega okolja za projekt Processing [30] in Wiring [31]. Zasnovan je bil predvsem tako, da omogoča relativno enostavno pisanje in nalaganje programov na Arduino ploščico, kar pomeni, da predstavlja zelo primerno izbiro za začetnike [32]. Slika 4.1 predstavlja primer napisanega programa v razvojnem okolju Arduino IDE.

---

<sup>1</sup>Arduino IDE: <http://arduino.cc/en/main/software>



Slika 4.1: Razvojno okolje Arduino IDE ter primer programa Blink.

Arduino IDE razvojno okolje je sicer zelo primerno za krajše programe, vendar pa se, ko želimo narediti nekoliko bolj kompleksen sistem oziroma ko želimo imeti v projektu več kot eno datoteko z izvirno kodo, pokažejo njegove pomanjkljivosti. Težave se začnejo pojavljati tudi, ko želimo izvajati nekoliko bolj podrobno analizo izvajanja napisanega programa. Ker je naš sistem bolj kompleksen oziroma potrebuje več izvirne kode, smo se v našem primeru zaradi lažjega razvijanja in testiranja odločili za razširitev Visual Micro<sup>2</sup> za razvojno okolje Microsoft Visual Studio. Visual Micro v osnovi uporablja Arduino IDE oziroma dodaja samo še en dodatni nivo abstraktnosti nad Arduino IDE, s čimer nam omogoča nekaj več dodatnih funkcij, ki olajšajo razvijanje bolj kompleksnih sistemov. Visual Micro predstavlja razširitev za razvojno okolje Microsoft Visual Studio, kar pomeni, da ni samostojen in tako moramo imeti nameščeno tudi orodje Microsoft Visual Studio. Microsoft Visual Studio predstavlja zelo močno razvojno okolje podjetja Microsoft, ki obstaja že od leta 1995 naprej in se neprestano nadgrajuje oziroma izpopolnjuje [33]. Razširitev Visual Mikro tako v bistvu predstavlja vmesni most med orodjema Microsoft Visual Studio in Arduino IDE in nam omogoča, da izkoristimo skoraj polno moč razvojnega okolja Microsoft Visual Studio pri razvoju programov za Arduino platformo. Nekaj izmed glavnih prednosti, ki jih ponuja ta kombinacija in so se v našem primeru izkazale za izredno koristne, so [34]:

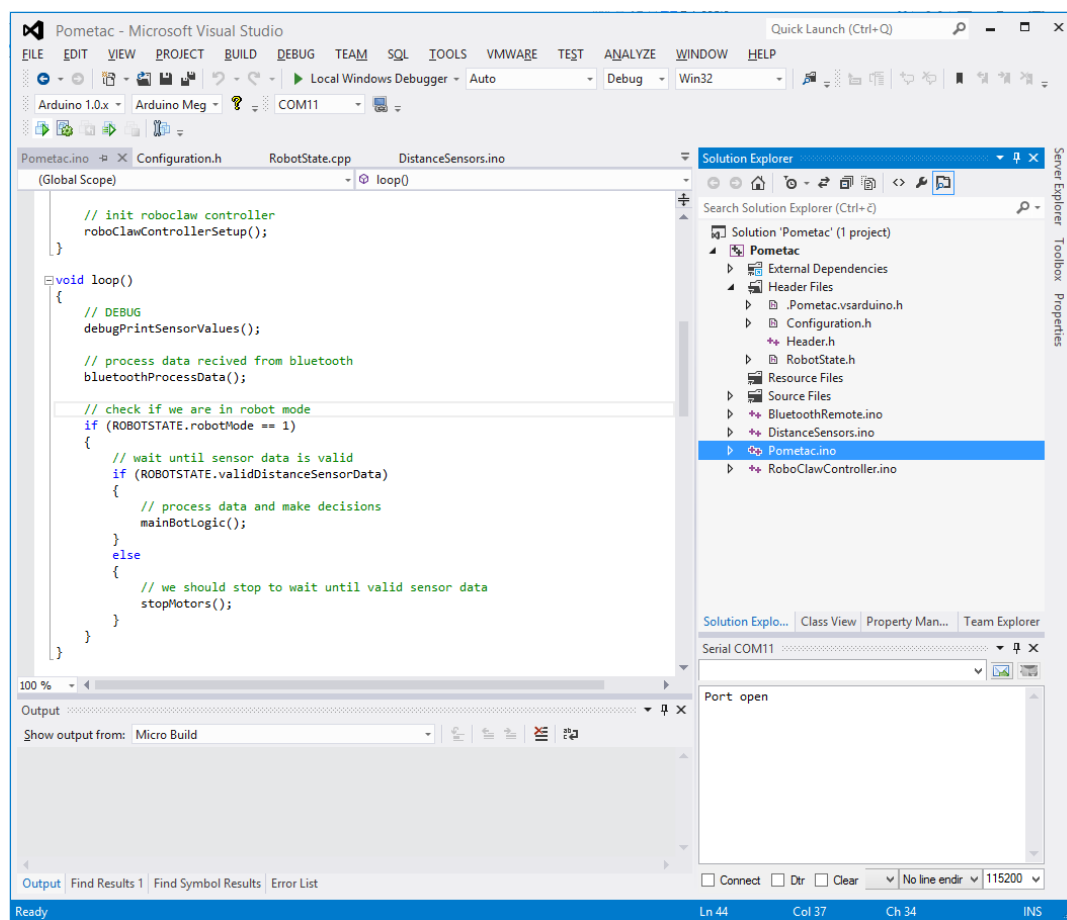
1. Omogoča napredni tekstovni urejevalnik z barvanjem izvirne kode in samodokončevanjem (angl. “autocomplete”).
2. Omogoča, da razbijemo izvirno kodo programa na več datotek.
3. Omogoča poglobljeno analizo izvajanja programa s pomočjo dodajanja prekinitvenih točk (angl. “breakpoints”) v programu ter vmesno analizo vsebine pomnilnika mikrokrmilnika Arduino.
4. Ohranja vso logiko prevajanja in prenašanja napisanega programa na mikrokrmilnik Arduino.

---

<sup>2</sup>Visual Micro razširitev: <http://www.visualmicro.com/>



Slika 4.2 predstavlja prikaz opisanega razvojnega okolja Microsoft Visual Studio z dodatno razširitvijo Visual Mikro, ki nam omogoča razvoj Arduino programov.



Slika 4.2: Uporabljeno razvojno okolje Microsoft Visual Studio z dodatno razširitvijo Visual Mikro, ter primer izvorne kode glavne zanke našega robota.

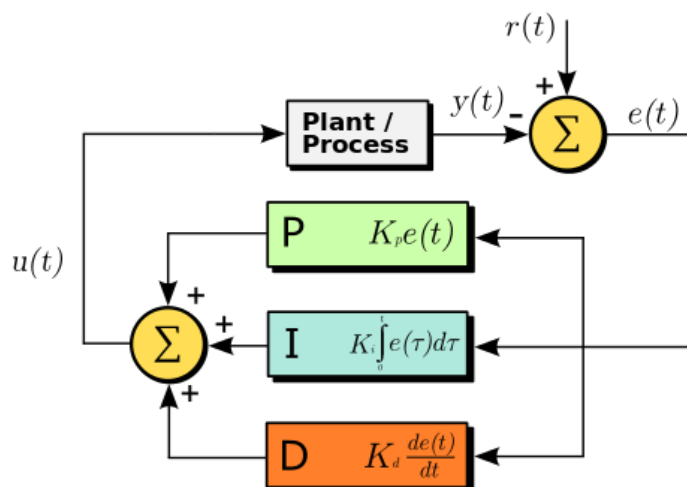
## 4.2 Premikanje robota

Podsistem premikanja robota v glavnem skrbi za krmiljenje motorjev ter magnetnih enkoderjev. Kot smo že omenili v Poglavlju 3.2, smo pri izgradnji pogonskega sistema robota uporabili še dodatno krmilno vezje RoboClaw podjetja Pololu za krmiljenje robotov motorja, ki je sposobno prenesti višje tokovne obremenitve motorja ter nam nekoliko poenostavi razvoj krmilne logike za nadzor motorjev. Na vezje RoboClaw se priklopijo magnetni enkoderji posameznega motorja, kjer ima vezje celo že implementirano logiko za ustrezno odčitavanje vrednosti teh enkoderjev. Tako je vse kar moramo implementirati na našem glavnem krmilnem vezju ustrezni vmesnik za to dodatno vezje RoboClaw. Tukaj smo se poslužili že napisane knjižnice<sup>3</sup> za Arduino, ki nam omogoča direktno dvosmerno serijsko komunikacijo z vezjem RoboClaw.

Prvi korak je bil, da smo morali najprej ustrezno konfigurirati oziroma nastaviti določene konstante vezja RoboClaw. In sicer smo morali nastaviti vrednosti P, I in D konstant PID kontrolorja [35], ki na podlagi teh konstant ter algoritma, predstavljenega na sliki 4.3 skrbi, da se odčitane vrednosti magnetnih enkoderjev prevedejo v pravilne vrednosti hitrosti in pospeška premikanja robota. P (angl. “proportional”) predstavlja sorazmernostno konstanto, I (angl. “integral”) predstavlja integralno konstanto ter D (angl. “derivative”) predstavlja derivatno konstanto. P, I in D konstante se določijo na podlagi uporabljenega motorja, ki je povezan na krmilno vezje. V našem primeru smo samo uporabili ustrezne vrednosti, ki jih je predlagalo podjetje Pololu, ki izdeluje motorje ter magnetne enkoderje, ki smo jih uporabili pri našem robotu. Nastaviti je potrebno še spodnjo mejo nivoja napetosti akumulatorja, pri kateri bo vezje RoboClaw izklopilo napajanje motorjev oziroma ustavilo njihovo delovanje.

---

<sup>3</sup>RoboClaw Arduino knjižnica: <http://downloads.orionrobotics.com/downloads/code/arduino.zip>



Slika 4.3: Diagram poteka algoritma PID krmilnika za izračun hitrosti ter pospeška premikanja robota [35].

Izbrali smo različne vrednosti hitrosti premikanja robota, ki predstavljajo ustrezen nivo napetosti motorjev. In sicer so to sledeče hitrosti:

1. “normal speed” – ta hitrost predstavlja skoraj maksimalno hitrost vrtenja motorjev in se uporablja ob normalnem delovanju oziroma takrat, ko v bližini robota ni nobene ovire. V našem primeru je ta hitrost približno 3 km/h (postopek izračuna maksimalne hitrosti robota se nahaja v Poglavju 3.2).
2. “creep speed” – ta hitrost vrtenja motorjev se uporablja takrat, ko sistem zazna, da je v bližini robota kakšna ovira oziroma da se je robot začel približevati kakšni oviri. Količinsko je ta hitrost približno ena tretjina vrednosti normalne hitrosti in se uporablja predvsem zato, da poskušamo zmanjšati verjetnost, da bi se robot pri polni hitrosti zaletel v kakšno oviro in bi pri tem lahko prišlo do kakšnih mehanskih poškodb. V našem primeru je ta hitrost približno 1 km/h.
3. “turn speed” – ta hitrost vrtenja motorjev se uporablja takrat, ko želimo robota obrniti bodisi v levo bodisi v desno smer.

4. “backwards speed” – ta hitrost vrtenja motorjev se uporablja takrat, ko se robot premika v vzvratno smer. V našem primeru je ta hitrost približno 3 km/h.

Poleg konfiguracije smo v tem podsistemu pripravili tudi vse ustrezne metode, ki omogočajo fizično premikanje robota v vse smeri. Prva oziroma najpomembnejša metoda je metoda, ki poskrbi, da se bo robot premikal v ravni liniji naravnost naprej. To metodo smo v izvorni kodi poimenovali “driveForward” in torej poskrbi, da se bo robot začel premikati v ravni liniji naravnost naprej z izbrano hitrostjo toliko časa, dokler ne pokličemo metode za ustavitev. V tej metodi je tudi implementirano branje vrednosti magnetnih enkoderjev z namenom, da zagotovimo, da se bosta oba motorja (levi in desni) premikala enako hitro. Če se motorja ne bosta premikala usklajeno, se robot ne bo premikal v ravni liniji, ampak se bo premikal v loku v levo ali desno smer. Torej, da lahko v tej metodi zagotovimo, da se bo robot premikal v ravni liniji, moramo neprestano brati hitrost vrtenja obeh motorjev preko omejene knjižnice RoboClaw ter na podlagi teh prebranih vrednosti hitrosti ustrezno prilagajati nivo napajanja motorjev. Za določanje oziroma prilagajanje vrednosti nivoja napetosti posameznega motorja smo uporabili algoritem proporcionalne regulacije [36]. In sicer ta algoritem v grobem deluje tako, da si izberemo en motor kot glavni (v našem primeru je to levi motor), ki bo imel vedno konstanto napetost, ter določimo drugi motor kot pomožni, kateremu se bo nivo napetosti spreminjala na podlagi njegove hitrosti v primerjavi s hitrostjo glavnega motorja. Nivo napetosti pomožnega motorja lahko izračunamo na podlagi formule (4.1), pri čemer smo morali še vnaprej določiti vrednost konstante  $K_p$ , ki se imenuje konstanta proporcionalnosti. Glavni namen te konstante je, da nekoliko ublaži kompenzacijo izračunane napake (razlike v hitrosti vrtenja glavnega in pomožnega motorja) oziroma da porazdeli kompenzacijo popravka nivoja napetosti pomožnega motorja čez nek specifični krajši časovni interval. Na podlagi testiranja smo določili, da je v našem primeru najbolj idealna vrednost konstante  $K_p$  enaka 30. Torej na podlagi opisane formule (4.1) neprestano preračunavamo nivo

napetosti pomožnega motorja in to vrednost pošljamo vezju RoboClaw preko knjižnice. Nivo napetosti glavnega motorja pa pošljemo samo enkrat, in sicer na začetku izvajanja te metode. Tako torej lahko zagotovimo, da se bo robot ob izvedbi te metode vedno premikal naprej v ravni liniji.

$$napetost_{pomozni} = napetost_{pomozni} + napaka \quad (4.1)$$

$$napaka = (hitrost_{glavni} - hitrost_{pomozni}) * K_p \quad (4.2)$$

Če pogledamo formulo za izračun napake (4.2), lahko ugotovimo, da če bi se motorja vrtela z enako hitrostjo, bi bila vrednost napake enaka 0. Če se pomožni motor vrti počasneje kot glavni, bi bila izračunana napaka pozitivna vrednost, če pa se pomožni motor vrti hitreje kot glavni, bi pa bila vrednost napake negativna.

Naslednja metoda, ki je implementirana v tem podsistemu, je metoda, ki poskrbi, da se bo robot premikal vzvratno. To metodo smo v naši izvorni kodi poimenovali “driveBackwards” in torej poskrbi, da se bo robot premikal v vzvratno smer z izbrano hitrostjo toliko časa, dokler ne pokličemo metode za ustavitev. Identično kot pri metodi za premikanje najprej tudi tukaj uporabljamo opisan algoritem proporcionalne regulacije, da lahko zagotovimo, da se bo robot premikal v ravni liniji nazaj oziroma da se ne bo premikal v levem ali desnem loku. V osnovi je ta metoda identična metodi “driveForward” le, da se pokličejo druge metode knjižnice RoboClaw, ki posrbijo, da se motorji vrtijo ravno v obratno smer.

Naslednji dve implementirani metodi sta metodi za obračanje robota “turnLeft” in “turnRight”. Ker tukaj ne potrebujemo tako velike natančnosti kot pri premikanju naprej/nazaj se algoritma proporcionalne regulacije ne poslužujemo, ampak samo preko RoboClaw knjižnice nastavimo ravno obratno smer vrtenja za posamezni motor z izbrano hitrostjo (prej definirano kot “turn speed”). Torej če želimo robota obrniti v levo, samo pošljemo ukaz desnemu motorju za premikanje naprej ter levemu motorju za premikanje

nazaj. Če želimo robota obrniti v desno, naredimo ravno obratno, torej levi motor naprej, desni nazaj.

Poleg metod za premikanje potrebujemo še metodo za ustavitev premikanja. To metodo smo v naši izvorni kodi poimenovali “stopMotors”, ki pošlje preko knjižnice ukaz vezju RoboClaw za takojšnjo popolno ustavitev obeh motorjev.

Kot zadnja metoda v tem podsistemu je metoda za detekcijo, ali robot stoji pri miru ali se premika. To metodo smo poimenovali “didStopedMoving” in v bistvu samo preveri, kdaj smo nazadnje prebrali pozitivno ali negativno vrednost hitrosti premikanja kateregakoli motorja. Če je ta razlika v času večja od nekega vnaprej določenega časovnega intervala (v našem primeru je to 1 sekunda) to pomeni, da se robot ne premika. Ta metoda je uporabna predvsem zaradi tega, da zaznamo primere, kot so na primer, če se je robot zataknil za nek predmet in ne more več naprej oziroma se je ustavil, nismo pa uspeli zaznati nobene ovire preko naših senzorjev.

### 4.3 Detekcija ovir

Podsistem za detekcijo ovir skrbi za ustrezno branje in interpretacijo vseh senzorjev, ki so namenjeni za detekcijo ovir. Prvi korak je najprej ustrezna inicializacija vseh senzorjev, za kar poskrbi metoda “distanceSensorsSetup”, ki torej poskrbi za ustrezno povezavo na posamezne senzorje ter jih resetira na začetno stanje. Naslednja in tudi najpomembnejša metoda v tem podsistemu je metoda “sensorsCheck”, ki ustrezno prebere vrednosti vseh senzorjev in izračuna, ali je podsistem zaznal oviro. V določenih primerih znamo celo določiti, koliko je ovira oddaljena od robota ter iz katere smeri se robotu približuje. Torej vsakič, ko pokličemo metodo “sensorsCheck” (ta metoda se kliče neprestano v zanki), ustrezno preverimo vse senzorje oziroma poskušamo določiti, ali smo zaznali oviro ter iz katere smeri se približuje robotu in kakšna je njena oddaljenost. Metoda “sensorsCheck” uporablja sledeče senzorje:

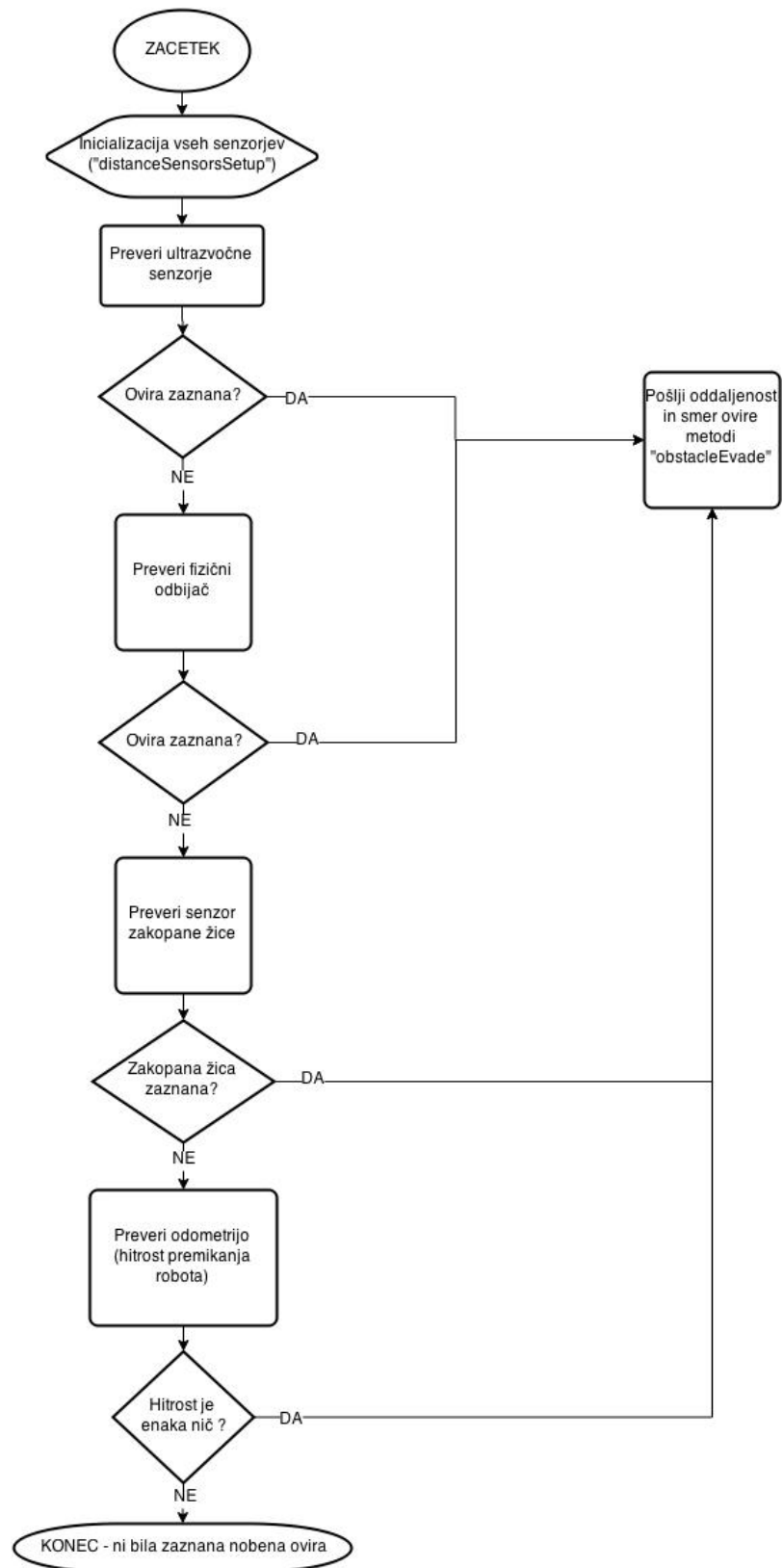
1. **Ultrazvočni senzorji** – predstavljajo prvi preventivni nivo detekcije ovir. Cilj teh senzorjev je, da poskušamo zaznati oviro, še preden se bo robot dejansko vanjo zaletel. V metodi “sensorsCheck” aktiviramo oddajnike vseh šest ultrazvočnih senzorjev ter nato na sprejemnikih poslušamo za povratni signal. Na podlagi časa, ki ga je potreboval signal, da je pripotoval nazaj do sprejemnika, lahko po opisanem postopku v Poglavju 3.6 izračunamo oddaljenost ovire od robota. Ker smo ultrazvočne senzorje enakomerno porazporedili na sprednjem delu robota, znamo tudi na podlagi informacije, na katerem sprejemniku smo zaznali povratni signal, določiti tudi smer, iz katere se robotu približuje ovira. Torej, če nam uspe zaznati oviro preko ultrazvočnih senzorjev, lahko metoda “sensorsCheck” vrne oddaljenost ovire ter smer, iz katere se ovira približuje robotu (največja razdalja, ki jo lahko zaznamo je 200 cm).
2. **Fizični odbijač** – predstavlja drugi nivo detekcije ovir, kar pomeni, da se je robot že zaletel v neko oviro oziroma jo z našimi ultrazvočnimi senzorji nismo uspeli zaznati. V metodi “sensorsCheck” samo enostavno za vse tri senzorje fizičnega odbijača (mikrostikala) preverimo, če je bilo katerokoli sproženo. Če je bilo katero sproženo, to pomeni, da se je robot zaletel v neko oviro. Ker imamo mikrostikala enakomerno porazporejena na sprednjem delu robota, lahko grobo tudi določimo, kje se nahaja ovira in ker se je robot že zaletel v oviro to pomeni, da je oddaljenost od ovire enaka nič. Torej, če zaznamo oviro preko senzorjev fizičnega odbijača, lahko metoda “sensorsCheck” vrne oddaljenost od ovire, ki je enaka nič, ter zazna približno smer, iz katere je prišla ovira.
3. **Senzor zakopane žice** – ta senzor se uporablja predvsem zato, da zaznamo, kdaj je robot prišel do meje njegovega območja delovanja in mora obrniti. Metoda “sensorsCheck” obravnava te nevidne meje območja delovanja enako, kot če bi se robot zaletel v neko oviro. Torej obravnava je enaka, kot če bi se robot zaletel na primer v steno, le da

se je v tem primeru zaletel v nevidno steno. Skratka, če zaznamo to nevidno mejo preko senzorjev zakopane žice, metoda “sensorsCheck” vrne oddaljenost od ovire, ki je enaka nič. Kar se tiče smeri, iz katere se je robot približal tej nevidni meji, pa jo v tem primeru ne znamo določiti, ker imamo senzorje zakopane žice nameščene samo na prednjem delu robota oziroma jih nimamo enakomerno razporejene, kot imamo to narejeno pri fizičnem odbijaču ali pa ultrazvočnih senzorjih.

4. **Odometrija** – ta senzor predstavlja zadnji nivo detekcije ovir. Saj enostavno samo preverimo trenutno hitrost vrtenja motorjev, in če zaznamo, da se motorji sploh ne vrtijo, vendar bi se morali, to pomeni, da je robot naletel na neko oviro, ki jo nismo bili sposobni zaznati z nobenim drugim senzorjem oziroma so vsi ostali senzorji odpovedali. V tem primeru metoda “sensorsCheck” vrne oddaljenost od ovire, ki je enaka nič oziroma vrne signal, da smo zaznali neko oviro. Kar se tiče smeri, iz katere je prišla ovira, jo pa v tem primeru prav tako ne znamo določiti.

Torej na podlagi zgoraj opisanih postopkov in korakov lahko z relativno veliko zanesljivostjo zaznamo katerokoli oviro, ki bi jo robot lahko srečal na nekem običajnem dvorišču. Slika 4.4 predstavlja diagram poteka opisanega algoritma detekcije ovir.





Slika 4.4: Diagram poteka algoritma zaznavanja ovir.

## 4.4 Obravnava ovir

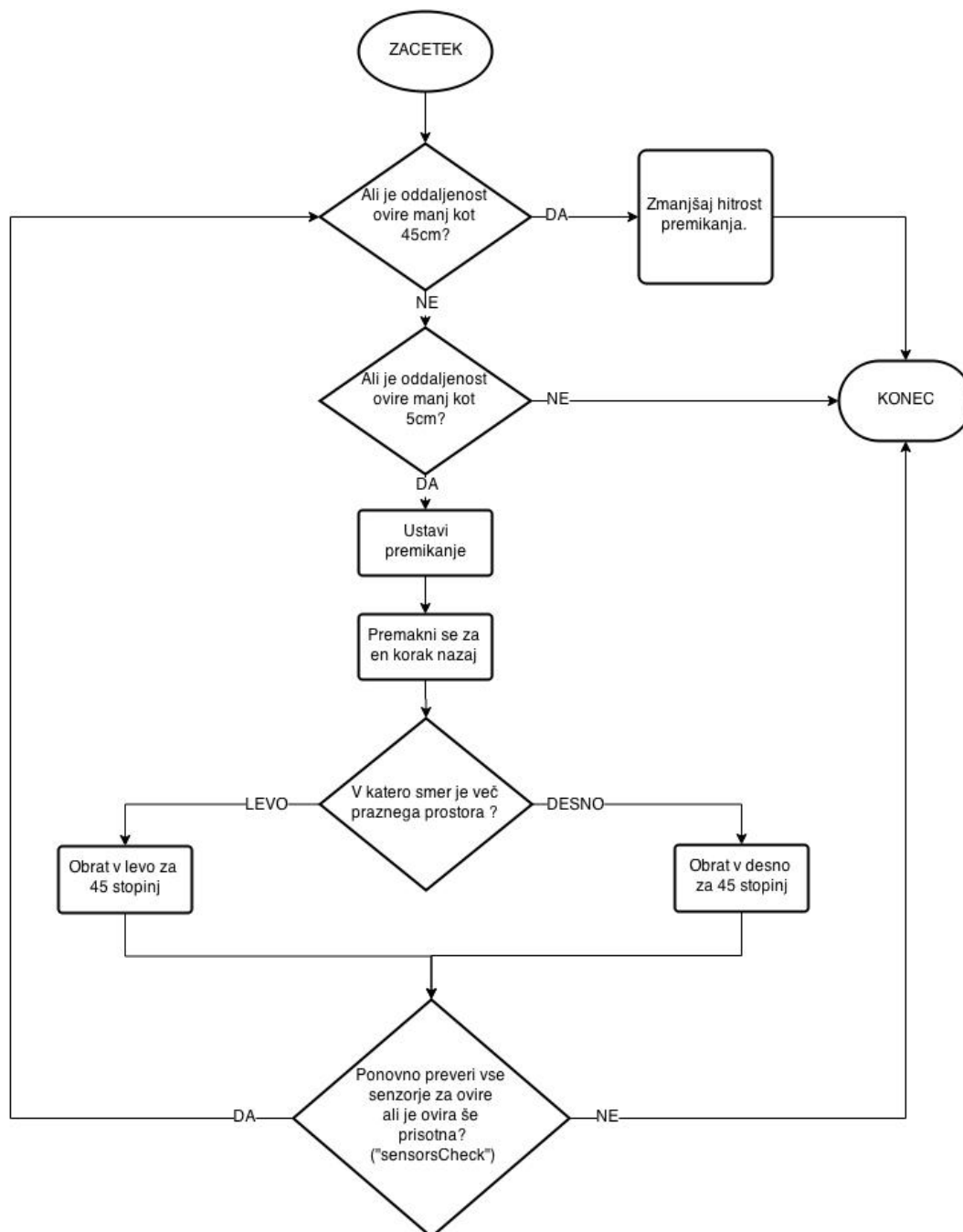
V prejšnjem poglavju smo opisali metodo “sensorsCheck”, ki nam določi, ali je bila zaznana kakšna ovira ter na kakšni oddaljenosti od robota je ta ovira in iz katere smeri se ovira robotu približuje. V tem poglavju bomo opisali odziv robota na te pridobljene informacije. Torej podsistem za obravnavo ovir neprestano kliče metodo “sensorsCheck” oziroma izvaja preverjanje prisotnosti ovire. Ko dobimo informacijo, da smo zaznali neko oviro, najprej preverimo, koliko je ta ovira oddaljena od robota. Če je oddaljena več kot 45 centimetrov, takoj znižamo hitrost premikanja robota na približno tretjino normalne hitrosti. S tem dosežemo, da se bo robot oviri približal počasi, kar zmanjša verjetnost, da bo prišlo do mehanskih poškodb robota, če se bo robot slučajno v oviro zaletel. Prav tako nam ta način tudi omogoča, da se oviri lahko bistveno bolj natančno približamo, kot pa če bi se robot premikal pri polni hitrosti. Če metoda “sensorsCheck” vrne oddaljenost od ovire, ki je manjša kot 5 centimetrov, pa se robot nemudoma ustavi ter začne izvajati metodo “obstacleEvade”. Metoda “obstacleEvade” je pa sestavljena iz več korakov:

1. Robot se najprej premakne za en korak nazaj, kar je približno 40 cm. To pa dosežemo tako, da preko podsistema za premikanje robota pošljemo ukaz za premikanje motorjev v vzvratno smer (izvedemo metodo “driveBackwards”) z ustrežno hitrostjo ter počakamo točno 0,5 sekunde ter nato pošljemo ukaz za ustavitev motorjev (metoda “stopMotors”). Če pomnožimo čas 0,5 sekunde z našo prej definirano hitrostjo vzvratnega premikanja (konstanta “backwards speed” definirana v poglavju o premikanju robota), ki je enaka 3 km/h, dobimo omenjeno razdaljo 40 cm, kar za nas predstavlja en korak.
2. Naslednji korak je, da določimo, v katero smer se bo robot obrnil. To pa naredimo tako, da zopet kličemo metodo “sensorsCheck” ter preverimo, v kateri smeri je največ prostega prostora (levo ali desno). Da lahko to določimo, enostavno samo seštejemo vse vrednosti razdalj

ultrazvočnih senzorjev, ki so obrnjeni v levo smer, ter seštejemo vse vrednosti razdalj ultrazvočnih senzorjev, ki so obrnjeni v desno smer, nato pa primerjamo ti dve vrednosti in tista, ki je manjša, v tisto smer se bo robot obrnil. Če pa sta vrednosti enaki pa se vedno odločimo za obrat v levo.

3. Potem, ko smo določili, v katero smer se bo robot obrnil, moramo ta obrat še dejansko izvesti. Obrat izvedemo tako, da pošljemo ukaz “turnLeft” oziroma “turnRight” na podsistem za premikanje robota, ki nato to ustrezno prevede v ukaze za motorje. Podobno kot pri prvem koraku tudi tukaj najprej pošljemo ukaz za začetek obračanja v izbrano smer (“turnLeft”/“turnRight”), počakamo točno 1,2 sekunde ter nato pošljemo ukaz za ustavitev motorjev (metoda “stopMotors”). Če upoštevamo definirano konstantno hitrost vrtenja motorjev, ko izvajamo obrat (3 km/h), ter medosno razdaljo našega robota (50 cm), lahko izračunamo, da smo s tem izvedli obrat za približno 45 °.
4. Kot zadnji korak po tem, ko smo izvedli obrat, pa robot samo še enkrat preveri vse senzorje preko metode “sensorsCheck”, če je kje še kakšna ovira, preden zopet preide v stanje premikanja naprej v ravni liniji. Če pa metoda “sensorsCheck” vrne še kakšno oviro, robot zopet ponovi celoten postopek obravnave ovir.

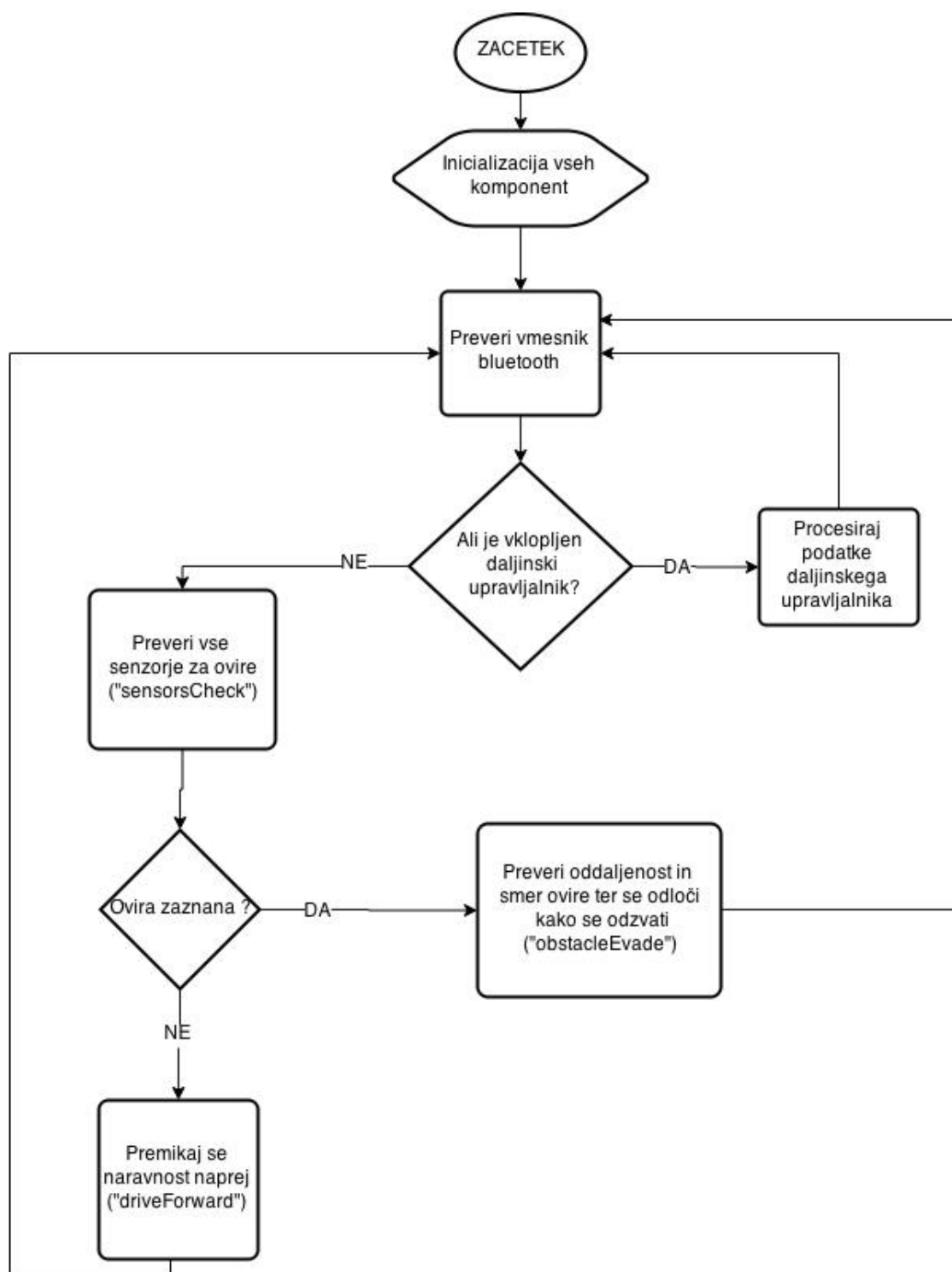
Slika 4.5 predstavlja diagram poteka opisanega algoritma obravnave ovir.



Slika 4.5: Diagram poteka algoritma obravnave ovir ("obstacleEvade").

## 4.5 Naključno premikanje

Podobno kot robotski sesalniki ter robotske kosilnice tudi naš robot deluje po principu naključnega premikanja. Torej naš robot temelji na principu, da če se dovolj dolgo naključno premikamo po nekem omejenem prostoru, bomo na tak način po določenem času preiskali večino prostora. Glavna krmilna metoda robota, ki smo jo v našem sistemu poimenovali “mainBotLogic”, torej zagotavlja naključno premikanje robota in povezuje vse prej opisane podsisteme v zaključeno celoto. Metoda vsebuje neskončno zanko, v kateri najprej preverimo, če se na vmesniku bluetooth nahaja kak podatek oziroma ukaz. Če zaznamo, da se nahaja kak ukaz, ga ustrezno obravnavamo oziroma preklopimo robota v poseben način delovanja, imenovan “remote control mode”, ki omogoča daljinsko upravljanje robota. Ta način delovanja je opisan v naslednjem poglavju. Če pa ne zaznamo nobenega ukaza na vmesniku bluetooth, pa nato preverimo preko metode “sensorsCheck”, če se robot slučajno približuje kakšni oviri. Če robot zazna oviro, se izvede metoda za obravnavo ovir (“obstacleEvade”), ki omogoča, da robot ustrezno odreagira na oviro ter se obrne v tisto smer, kjer lahko nadaljuje svojo pot naravnost naprej. Če pa robot ne zazna nobene ovire, se izvede metoda “driveForward” oziroma se robot premika v ravni liniji naprej. Opisan algoritem je s pomočjo diagrama poteka predstavljen na sliki 4.6.



Slika 4.6: Diagram poteka naključnega premikanja robota oziroma glavne krmilne metode robota.

## 4.6 Oddaljen nadzor preko vmesnika bluetooth

Kot dodaten modul smo našemu robotu dodali še podsistem za oddaljen nadzor preko bluetooth povezave. In sicer smo izkoristili obstoječo brezplačno aplikacijo za android pametne telefone, imenovano BlueBots<sup>4</sup>, ki služi kot daljinski upravljalnik za našega robota.

Aplikacija nam torej omogoča, da lahko premikamo robota s pomočjo simulirane krmilne palice ali pa celo s pomočjo pospeškometra telefona oziroma tako, da nagibamo telefon v različne smeri. Aplikacije te nagibe telefona oziroma premike krmilne palice najprej ustrezno pretvori v kartezični koordinatni sistem (izračuna X in Y koordinatne vrednosti), ter jih nato preko bluetooth povezave pošilja k našemu robotu, kjer jih potem ustrezno pretvorimo v napetostne nivoje za oba motorja. Na samem robotu imamo nameščen dodaten bluetooth modul, ki omogoča dvosmerno komunikacijo s poljubno napravo, ki podpira bluetooth standard.

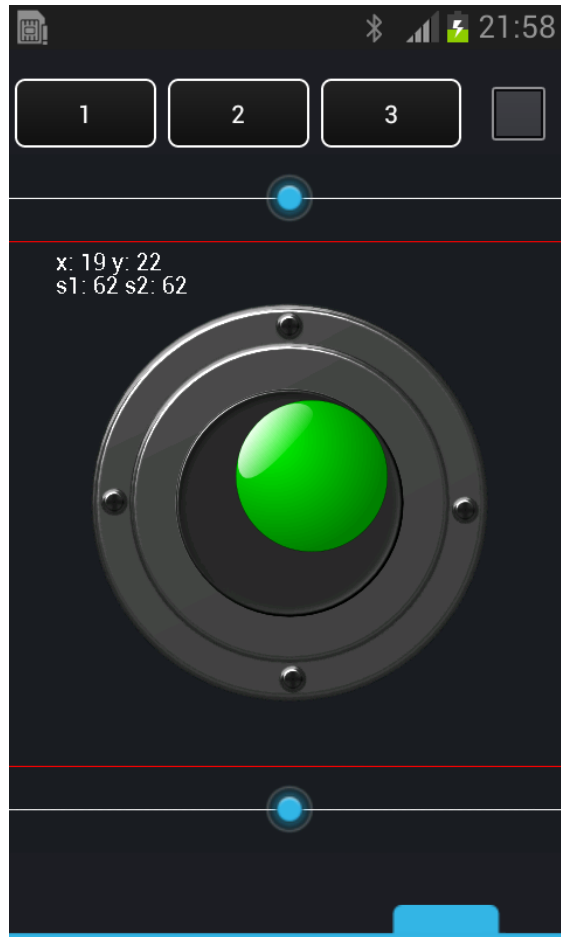
V naši glavni krmilni metodi najprej ustrezno inicializiramo ta modul, nato pa začnemo poslušati morebitne povezave. Če zaznamo, da je bila vzpostavljena nova bluetooth povezava, prekinemo v prejšnjem poglavju opisano metodo naključnega premikanja, ter začnemo poslušati ter izvajati ukaze, prejete preko tega bluetooth modula. Aplikacija BlueBots periodično v paketih pošilja robotu X in Y koordinatne vrednosti krmilne palice, ki jih moramo še ustrezno pretvoriti v napetostne nivoje obeh motorjev, zato da se bo robot dejansko premikal skladno s premiki kontrolne palice. Napetostni nivo posameznega motorja izračunamo na podlagi formule (4.3). Te izračunane napetostnega nivoja pošljemo na oba motorja preko RoboClaw knjižnice, pri čemer negativne vrednosti napetostnega nivoja pomenijo vrtenje motorja v ravno obratno smer. S tem v bistvu zagotovimo, da se bo robot premikal skladno s premiki krmilne palice na pametnem telefonu oziroma skladno

---

<sup>4</sup>Aplikacija BlueBots: <https://play.google.com/store/apps/details?id=blue.bots.free.matt>

z nagibom pametnega telefona, kar nam torej omogoča daljinsko upravljanje oziroma daljinsko premikanje robota. Slika 4.7 predstavlja ponazoritev uporabniškega vmesnika uporabljene aplikacije BlueBot oziroma opisano simulirano krmilno palico.

$$\begin{aligned} NAPETOST_{Levi-motor} &= Y + X \\ NAPETOST_{Desni-motor} &= Y - X \end{aligned} \quad (4.3)$$



Slika 4.7: Uporabniški vmesnik aplikacije BlueBot oziroma uporabljena simulirana krmilna palica za daljinsko upravljanje robota.



## Poglavje 5

# Analiza delovanja robota

Kot smo že nekajkrat omenili, je bil naš cilj narediti robota, ki bi bil sposoben relativno učinkovito pomesti običajno dvorišče. Slike 5.1, 5.2, 5.3 in 5.4 prikazujejo končni izdelek, slikan iz različnih kotov. V tem poglavju bomo predstavili rezultate analize delovanja robota. Izvedli smo kar nekaj meritev ter za demonstracijo poskusov posneli tudi nekaj video posnetkov različnih primerov, ki jih bomo opisali v nadaljevanju tega poglavja. Ker je zunanje okolje relativno kompleksno oziroma obstaja ogromno različnih spremenljivk, se bomo v tem diplomskem delu omejili le na eno testno dvorišče oziroma bomo omejili testiranje ter analizo delovanja le na to dvorišče.



Slika 5.1: Končni izdelek.



Slika 5.2: Končni izdelek.





Slika 5.3: Končni izdelek.



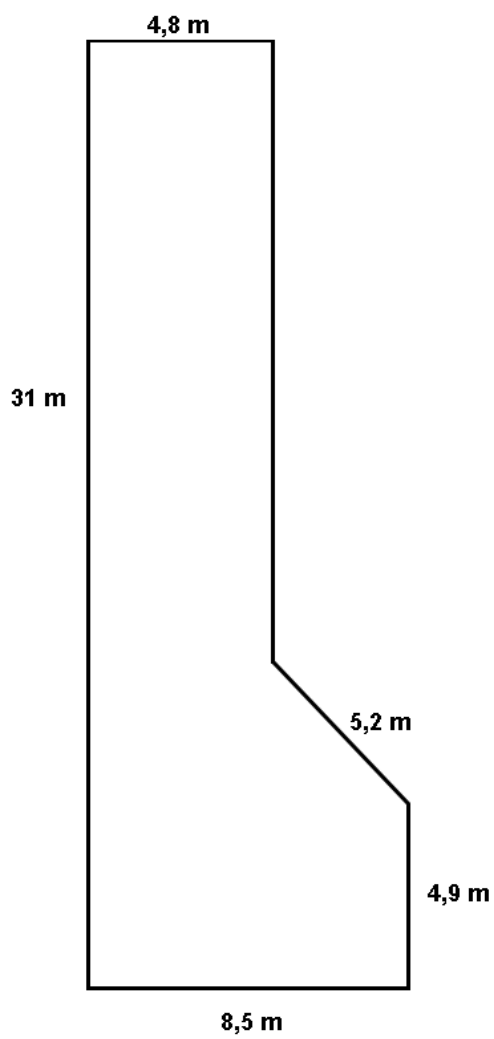
Slika 5.4: Končni izdelek.

## 5.1 Testno dvorišče

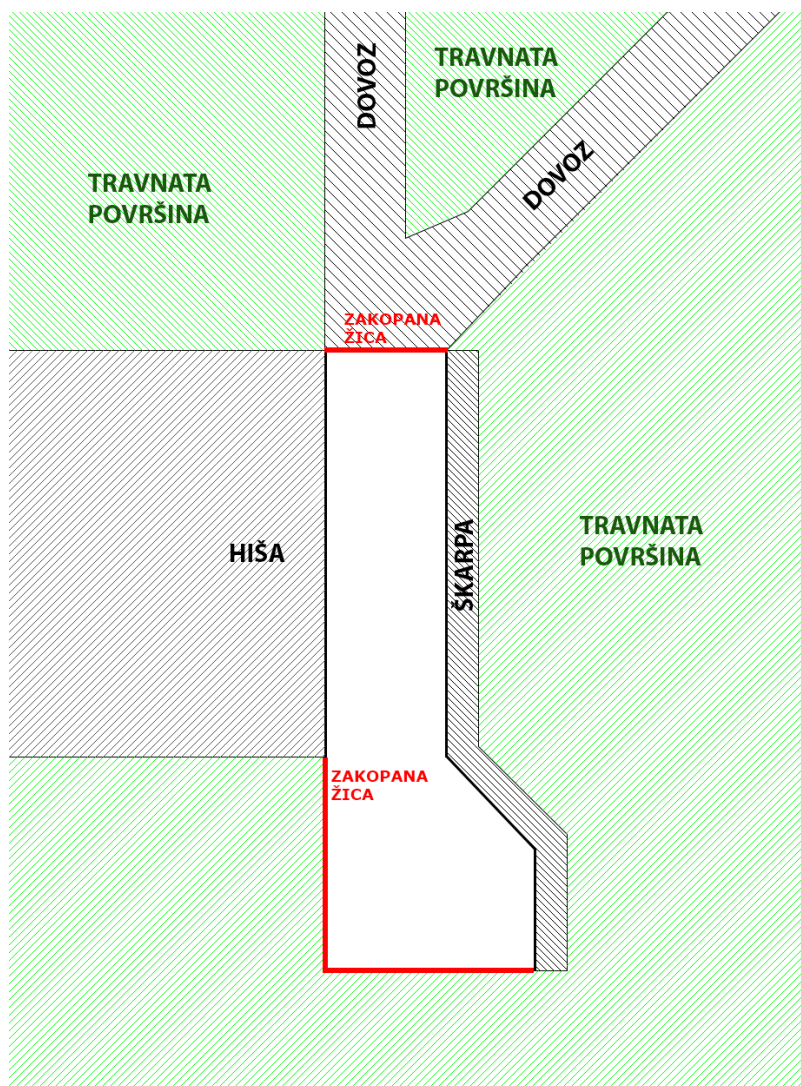
Dvorišče, na katerem smo testirali oziroma analizirali delovanje, ima obliko nekakšnega škornja ter ima skupno površino približno 176 kvadratnih metrov. Slika 5.5 ponazarja točno obliko dvorišča ter njegove dimenzije, pri čemer dvorišče meji s sledečimi objekti:

1. Zunanjo steno hiše,
2. približno meter visoko betonsko škarko,
3. travnato površino,
4. dovozom, ki vsebuje rahel klanec.

Slika 5.6 ponazarja skico dvorišča, pri čemer so označeni vsi omejeni objekti, s katerimi dvorišče meji. Torej, robot z zunanjimi stenami in betonsko škarko nima večjih težav, saj jih lahko relativno zanesljivo zazna z ultrazvočnimi senzorji. Nekaj več problemov pa ima robot s travnato površino ter dovozom, ker tega ultrazvočni senzorji niso sposobni zaznati, zato smo morali na meji s temi objekti položiti zakopano žico in tako omejiti območje delovanja robota. Na sliki 5.6 smo točno označili, kako smo položili zakopano žico, pri čemer smo vsega skupaj porabili približno 22 metrov zakopane žice. Slika 5.7 pa predstavlja še fotografsko ponazoritev testnega dvorišča. Sedaj, ko smo robotu točno določili njegovo območje delovanja, pa ostane samo še, da ga postavimo znotraj definirane območja ter ga vključimo. Robot takoj začne izvajati algoritem naključnega premikanja oziroma se naključno premika toliko časa dokler ga ne ustavimo ali pa mu zmanjka električne energije.



Slika 5.5: Oblika ter dimenzije testnega dvorišča.



Slika 5.6: Skica testnega dvorišča z označeno zakopano žico.

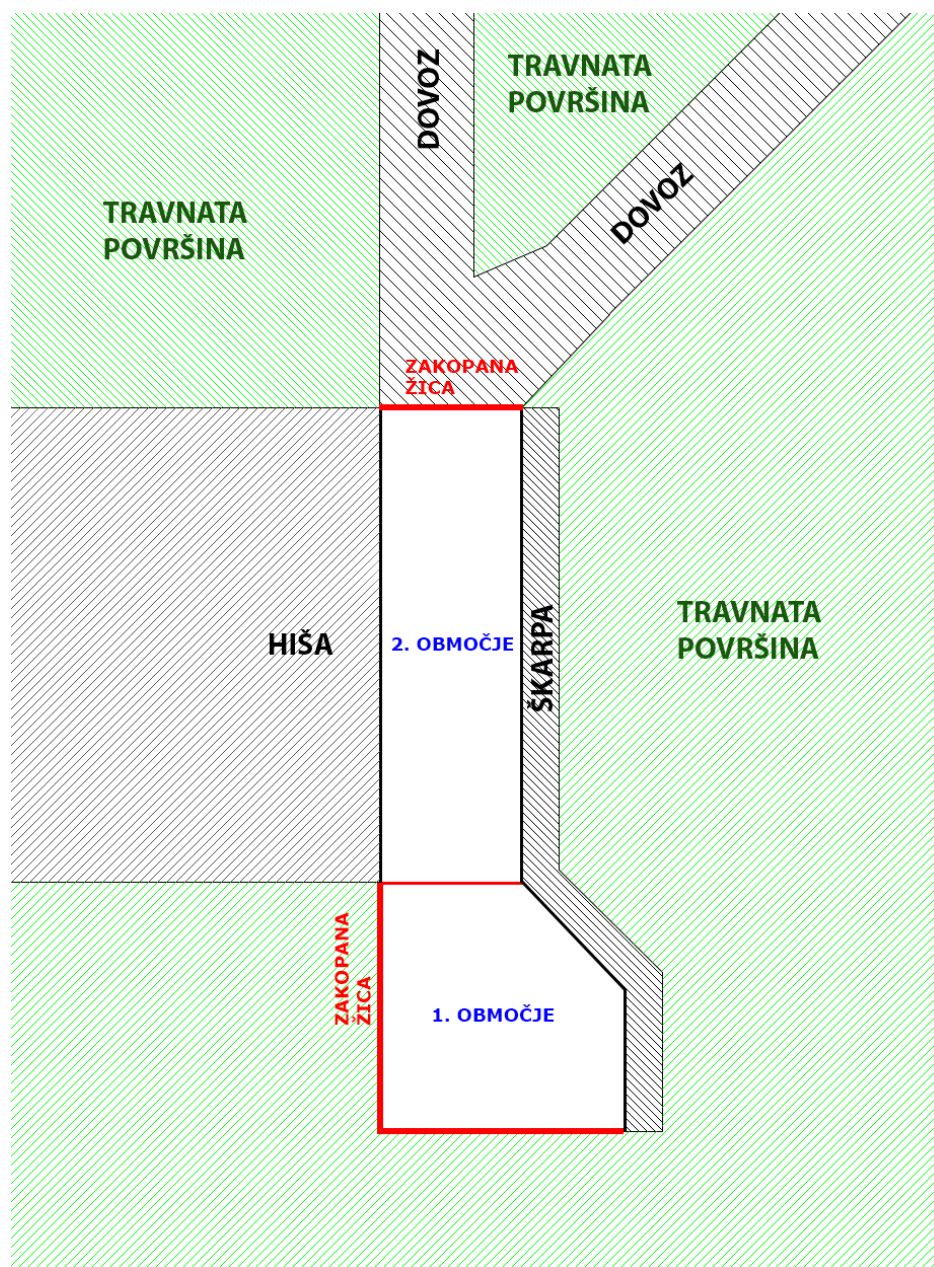


Slika 5.7: Fotografija testnega dvorišča.

## 5.2 Analiza obiskanosti območja delovanja

Analizo obiskanosti območja delovanja smo opravili tako, da smo na balkon hiše namestili kamero visoke ločljivosti ter posneli delovanje robota z neke višje točke. Ker balkon oziroma ta točka namestitve kamere ni dovolj visoka, da bi lahko naenkrat zajeli celotno testno dvorišče, smo morali razbiti testno dvorišče na dva dela oziroma pripraviti dva ločena posnetka. Kamero smo pritrdili na ograjo balkona s pomočjo stojala ter jo obrnili tako, da smo poskušali v obeh primerih zajeti čim večji del testnega dvorišča. Ker smo imeli na voljo samo eno kamero visoke ločljivosti, smo najprej posneli prvo območje ter nato kamero premaknili na novo lokacijo ter ponovili poskus še za drugo območje. Območja smo med seboj ločili s pomočjo dodatne zakopane žice, kar nam omogoča da omejimo delovanje robota na notranjost posameznega testnega območja dvorišča. Slika 5.8 prikazuje opisano razdelitev dvorišča na dva testna območja. Slika 5.9 predstavlja izsek s posnetka prvega testnega območja, ki je veliko približno 68 kvadratnih metrov. Slika 5.10 pa predstavlja izsek s posnetka drugega testnega območja, ki je veliko približno 88 kvadratnih metrov.





Slika 5.8: Skica definiranih območji delovanja za analizo obiskanosti.



Slika 5.9: Slika obratovanja na prvem testnem območju.



Slika 5.10: Slika obratovanja na drugem testnem območju.

Ko imamo enkrat pripravljene video posnetke delovanja znotraj posameznega območja, je naslednji korak izračun, kolikšen odstotek tega območja je robot obiskal v določenem času. To smo pa storili tako, da smo najprej iz posameznega video posnetka vzeli sliko praznega dvorišča ter nato za vsako nadaljnjo sliko video posnetka, ko se na njej nahaja robot, izračunali razliko v primerjavi s to sliko praznega dvorišča. Ta razlika med sliko praznega dvorišča in sliko dvorišča, na kateri se nahaja robot, pomeni ravno pozicija robota oziroma kolikšen del območja je robot obiskal v tistem trenutku, pod pogojem, da se kamera ne premika. Če to naredimo za vse posamezne slike video posnetka, lahko izračunamo, kolikšen delež definiranega območja je robot dejansko obiskal. Tabela 5.1 predstavlja izračunane rezultate obiskanosti posameznega testnega območja.

Testno območje	Obiskanost	Čas
1. območje	84 %	29 min
2. območje	89 %	38 min

Tabela 5.1: Rezultati izračunane obiskanosti posameznega testnega območja.

Na podlagi tabele rezultatov 5.1 lahko izračunamo, da ima robot na prvem testnem območju sposobnost delovanja približno 2,4 kvadratnih metrov na minuto, na drugem testnem območju pa 2,3 kvadratnih metrov na minuto. Če izračunamo povprečje, dobimo sposobnost delovanja približno 2,35 kvadratnih metrov na minuto. Na podlagi tabele 5.1 lahko izračunamo tudi povprečno obiskanost, ki znaša 86 %. Torej na podlagi teh izračunov oziroma rezultatov lahko sklepamo, da bi za celotno dvorišče, velikosti 176 kvadratnih metrov, robot porabil 75 minut za obiskanost približno 86 %.

Zaradi perspektivne projekcije kamere te dobljeni oziroma izračunani rezultati predstavljajo le grobo oceno obiskanosti dvorišča. Prav tako obstaja določeno območje pod balkonom, na katerem je bila nameščena kamera, ki ga zaradi pozicije kamere v nobenem primeru nismo mogli zajeti oziroma ni

bilo vključeno v opisano testiranje. To območje je veliko približno 20 kvadratnih metrov in za njega sklepamo, da bo imel približno podobno povprečno obiskanost kot prej opisana območja.

### 5.3 Analiza zaznavanja ovir

V tem poglavju bomo predstavili rezultate analize zaznavanja različnih tipov ovir, ki se nahajajo na našem testnem dvorišču. Uporabili smo ovire, ki bi se lahko nahajale tudi na drugih podobnih dvoriščih. V vseh poskusih smo robota postavili približno 2 metra neposredno pred obravnavano oviro ter ga vklopili. Robot se oviri približa ter jo zazna bodisi s pomočjo ultrazvočnih senzorjev bodisi s pomočjo fizičnega odbijača. Če je robot pred oviro pravočasno zmanjšal hitrost ter se ustavil preden se je dejansko fizično dotaknil ovire, to pomeni, da je oviro zaznal s pomočjo ultrazvočnih senzorjev. Če pa se je robot zaletel v oviro, ter šele nato ustavil oziroma premaknil za en korak nazaj, pa to pomeni, da je oviro zaznal s pomočjo fizičnega odbijača. Pri vsaki oviri smo opravili več poskusov ter prešteli, v koliko primerih je robot oviro zaznal s pomočjo ultrazvočnih senzorjev ter v koliko primerih s pomočjo fizičnega odbijača, ter nato te rezultate predstavili v tabeli. Za lažjo predstavbo smo tudi za vsako oviro pripravili video posnetek, ki prikazuje primer opravljenega poskusa. Analizirali smo zaznavanje naslednjih ovir:

1. **Stene hiše ter betonska škarpa** – predstavljajo oviro, na katero bo robot na našem testnem dvorišču največkrat naletel. Analizo zaznave te ovire smo opravili po prej opisanem postopku štetja zaznav z različnimi senzorji, pri čemer smo vsakič robota proti steni oziroma škarpri poslali pod nekoliko različnim vpadnim kotom ter za vsak vpadni kot opravili 10 ponovitev (5 za betonsko škarpno ter 5 za steno hiše). Rezultate poskusov smo zabeležili v tabeli 5.2. Na podlagi teh rezultatov lahko izračunamo, da je uspešnost zaznave obravnavane ovire s pomočjo ultrazvočnih senzorjev 87,5 %. Slika 5.11 predstavlja slikovni



izsek pripravljenega video posnetka poskusa <sup>1</sup>.

Vpadni kot	Št. vseh poskusov	Št. zaznav s pomočjo ultrazvočnih senzorjev	Št. zaznav s pomočjo fizičnega odbijača
0°	10	10	0
20°	10	8	2
45°	10	9	1
60°	10	8	2

Tabela 5.2: Rezultati poskusov zaznavanja stene hiše ter betonske škarpe.



Slika 5.11: Primer poskusa zaznavanja ter obravnavanja sten hiše pri vpadnem kotu 0°.

<sup>1</sup>Obravnavna sten hiše ter betonske škarpe - <http://youtu.be/rZzhrNAXwcU>

2. **Zakopana žica** – s pomočjo zakopane žice smo robotu omejili območje delovanja tam, kjer ni nobenega drugega mejnega predmeta, ki bi ga lahko zaznali z pomočjo ultrazvočnih senzorjev ali fizičnega odbijača. Na našem testnem dvorišču je to dovoz na dvorišče ter travnata površina. Analizo zaznavanja zakopane žice smo opravili na podoben način, kot je bil na začetku tega poglavja opisan postopek štetja zaznav z različnimi senzorji. Razlika je ta, da smo namesto štetja zaznav s pomočjo ultrazvočnih senzorjev ter fizičnega odbijača, šteli zaznave s pomočjo senzorja zakopane žice. Podobno kot v prejšnjem primeru smo tudi tukaj robota poslali proti zakopani žici vsakič pod nekoliko različnim vpadnim kotom ter za vsak vpadni kot opravili 10 ponovitev poskusa. Rezultate smo zabeležili v tabeli 5.3. Na podlagi teh rezultatov lahko izračunamo, da je uspešnost zaznave zakopane žice 95 %. Slika 5.12 predstavlja slikovni izsek pripravljenega video posnetka poskusa <sup>2</sup>.

Vpadni kot	Št. vseh poskusov	Št. uspešnih zaznav
0 °	10	10
20 °	10	9
45 °	10	9
60 °	10	10

Tabela 5.3: Rezultati poskusov zaznavanja zakopane žice.

<sup>2</sup>Obravnava zakopane žice - [http://youtu.be/Qp\\_jUD-CroU](http://youtu.be/Qp_jUD-CroU)



Slika 5.12: Primer poskusa zaznavanja zakopane žice pri vpadnem kotu  $0^\circ$ .

3. **Statična ovira (lesena klop)** – lesena klop predstavlja enega izmed tipičnih predmetov, na katerega bi robot lahko naletel na nekem dvorišču. Ker ima klop relativno ozke noge, predstavlja to za robota nekoliko bolj kompleksno oviro. Na podlagi opisanega postopka štetja zaznav z različnimi senzorji smo opravili 10 ponovitev poskusa ter rezultate predstavili v tabeli 5.4. Na podlagi teh rezultatov lahko izračunamo, da je uspešnost zaznave obravnavane ovire s pomočjo ultrazvočnih senzorjev 30 %, kar je nekako pričakovano glede na obliko ovire. Slika 5.13 predstavlja slikovni izsek pripravljenega video posnetka poskusa <sup>3</sup>.

---

<sup>3</sup>Obravnava statične ovire (lesena klop) - <http://youtu.be/iEfppEJGnFQ>

Št. vseh poskusov	Št. zaznav s pomočjo ultrazvočnih senzorjev	Št. zaznav s pomočjo fizičnega odbijača
10	3	7

Tabela 5.4: Rezultati poskusov zaznavanja lesene klopi.



Slika 5.13: Primer poskusa zaznavanja lesene klopi.



4. **Statična ovira (betonska stopnica)** – na našem testnem dvorišču se pred vhodom v hišo nahaja betonska stopnica, višine približno 20 centimetrov. Ker je betonska stopnica pravokotne oblike to pomeni, da ima precej oster rob. Kot smo že v Poglavju 3.6 (Ultrasvočni senzorji) opisali, imajo ultrasvočni senzorji pri predmetih take oblike nekaj težav, saj se namreč ultrasvok odbije ravno v napačno smer in tako senzorji ne dobijo povratnega signala. Na podlagi opisanega postopka štetja zaznav z različnimi senzorji smo opravili 10 ponovitev poskusa ter rezultate zabeležili v tabeli 5.5. Na podlagi rezultatov je jasno razvidno, da so v tem primeru ultrasvočni senzorji povsem odpovedali. Slika 5.14 predstavlja slikovni izsek pripravljenega video posnetka poskusa <sup>4</sup>.

<b>Št. vseh poskusov</b>	<b>Št. zaznav s pomočjo ultrasvočnih senzorjev</b>	<b>Št. zaznav s pomočjo fizičnega odbijača</b>
10	0	10

Tabela 5.5: Rezultati poskusov zaznavanja roba betonske stopnice.

---

<sup>4</sup>Obravnava statične ovire (betonska stopnica) – <http://youtu.be/8beZyXtwhtg>



Slika 5.14: Primer poskusa zaznavanja roba betonske stopnice.

5. **Premična ovira (človek)** – kot primer premične ovire, na katero bi lahko robot naletel na običajnem dvorišču, smo uporabili človeka, ki se robotu ni uspel pravočasno umakniti. Podobno kot v prejšnjih primerih smo tudi tukaj uporabili opisan postopek štetja zaznav z različnimi senzorji ter rezultate poskusov predstavili v tabeli 5.6. Na podlagi teh rezultatov lahko izračunamo, da je uspešnost zaznave obravnavane ovire s pomočjo ultrazvočnih senzorjev 60 %. Slika 5.15 predstavlja slikovni izsek pripravljenega video posnetka poskusa <sup>5</sup>.

---

<sup>5</sup>Obravnava premične ovire (človek) - [http://youtu.be/W8JAsFU\\_15Y](http://youtu.be/W8JAsFU_15Y)

Št. vseh poskusov	Št. zaznav s pomočjo ultrazvočnih senzorjev	Št. zaznav s pomočjo fizičnega odbijača
10	6	4

Tabela 5.6: Rezultati poskusov zaznavanja premične ovire - človeka.



Slika 5.15: Primer poskusa zaznavanja premične ovire - človeka.

## 5.4 Analiza daljinskega upravljalnika

Kot zadnjo analizo delovanja smo testirali odzivnost oziroma natančnost opisanega daljinskega upravljalnika (aplikacije na pametnem telefonu). Kot rezultat odzivnosti štejemo pretečen čas od trenutka, ko s telefona pošljemo ukaz, do trenutka, ko se začne robot premikati <sup>6</sup>. Tabela 5.7 predstavlja rezultate našega testiranja odzivnosti daljinskega upravljalnika. Kot je iz tabele rezultatov razvidno, so zakasnitve relativno minimalne in ne vplivajo na uporabnost našega daljinskega upravljalnika. Slika 5.16 predstavlja primer testiranja daljinskega upravljalnika (izsek iz videoposnetka).

Ukaz	Pretečen čas do izvedbe ukaza
Premik naprej	589ms
Premik nazaj	637ms
Obrat v levo	504ms
Obrat v desno	591ms

Tabela 5.7: Rezultati testiranja odzivnosti daljinskega upravljalnika.

---

<sup>6</sup>Premikanje s pomočjo daljinskega upravljalnika - <http://youtu.be/FrKyeX0yvIs>



Slika 5.16: Primer testiranja daljinskega upravljalnika.



## Poglavje 6

### Zaključek

Glavni cilj tega diplomskega dela je bil sestaviti ter ustrezno programirati robotskega pometača dvorišča, ki bo znal zadovoljivo dobro in avtonomno pomesti dvorišče. Najprej smo naredili analizo podobnih robotov, in sicer smo pogledali, kako so zgrajeni oziroma katere senzorje uporabljajo za kakšen namen, nato pa smo analizirali še njihove algoritme premikanja. Analizirali smo predvsem obstoječe rešitve robotskih sesalnikov ter robotskih kosilnic, iz katerih smo lahko potegnili marsikatero vzporednico z našim izzivom.

Nadaljevali smo z opisom vseh uporabljenih komponent našega robota. Ker smo si želeli delo nekoliko poenostaviti, smo za osnovo vzeli obstoječ ročni pometač podjetja Kärcher, ki smo ga nadgradili z električnim pogonskim sistemom ter različnimi senzorji. Pri namestitvi komponent na osnovo nam je tudi zelo prav prišla relativno nova tehnologija 3D tiskanja, saj smo si lahko marsikatero ohišje ali pa nosilec enostavno kar natisnili (na podlagi narisane 3D računalniškega modela).

Po opisu vseh uporabljenih komponent našega robota smo nadaljevali z opisom krmilne logike robota, ki zna pravilno interpretirati ter povezati skupaj podatke iz različnih senzorjev, na podlagi katerih lahko potem robot preiskuje svoje območje delovanja. V našem primeru smo se odločili za preiskovanje prostora oziroma območja delovanja na osnovi naključnega algoritma, ki temelji na principu, da če se robot v prostoru dovolj dolgo premika,

bo v določenem času obiskal večino prostora.

V zadnjem koraku tega diplomskega dela smo opisali postopek testiranja ter rezultate analize delovanja robota na našem testnem dvorišču. Za analizo obiskanosti smo razbili dvorišče na dve ločeni območji, ter za vsakega posebej posneli video posnetek delovanja robota, na podlagi katerega smo nato izračunali odstotek obiskanosti območja delovanja. Na podlagi rezultatov obiskanosti posameznega območja smo nato izpeljali povprečni čas ter odstotek obiskanosti za celotno testno dvorišče. Testirali ter analizirali smo tudi zaznavanje različnih tipov ovir, ter za vsak posamezni tip ovire izračunali odstotek uspešnosti zaznave z različnimi senzorji. Kot zadnjo analizo smo še preverili odzivnost daljinskega upravljalnika.

## 6.1 Predlog nadaljnjih izboljšav

Ker smo našo analizo delovanja robota omejili le na eno testno dvorišče, je prvi predlog nadaljnjega dela oziroma izboljšav razširitev analize na več različnih tipov dvorišč. Na podlagi te analize oziroma ugotovitev bi potem lahko ustrezno prilagodili krmilno logiko robota oziroma morebiti celo dodali še kakšne dodatne senzorje, če bi ugotovili, da je potreba po tem, oziroma da robot ne dobi dovolj podatkov iz okolja, da bi se lahko pravilno odločal.

Naslednja najbolj očitna možna izboljšava bi bila, da bi zamenjali senzorje z bolj zmogljivimi oziroma bolj natančnimi verzijami, vendar pa to pomeni, da bi sam strošek izdelave tudi narasel. Tukaj bi se tako torej morali odločiti za nek kompromis med ceno in zmogljivostjo senzorjev, ki bi bil za naš primer najbolj primeren.

Naslednja možna izboljšava našega robota bi bila, da bi dodali še senzor, ki bi zaznal, kdaj je zbiralnik za smeti poln in na podlagi katerega bi lahko razvili dodatno krmilno logiko, da bi se robot ustavil in z zvočnim signalom sporočil, da ima poln zbiralnik za smeti in s tem obvestil končnega uporabnika, da mora zbiralnik izprazniti. Še boljša izboljšava bi bila, da bi sestavili mehanizem za avtomatsko praznjenje zbiralnika oziroma da bi robot zaznal



kdaj ima poln zbiralnik za smeti ter ga šel nato avtomatsko izpraznil na neko izbrano mesto.

Kot še ena izmed možnih izboljšav našega robota bi lahko bila tudi, da bi vse senzorje oziroma vso elektroniko nekoliko bolje zaščitili proti vodi. Glede na to, da robot deluje v zunanjem okolju se načeloma lahko zgodi na primer, da bi zapeljal čez kakšno lužo ali pa na primer, da bi začelo deževati med samim delovanjem robota.

Naslednja možna izboljšava bi lahko bila izboljšanje algoritma naključnega premikanja ali pa celo implementacija načina premikanja na podlagi zgrajenega zemljevida. Hipotetično bi bilo možno zgraditi nek približen zemljevid na podlagi podatkov, pridobljenih iz obstoječih senzorjev, lahko bi pa dodali še dodatne funkcionalnosti, kot je na primer računalniški vid oziroma sistem stereo vida s pomočjo dveh kamer. Več kot bi imeli na voljo podatkov o okolju iz različnih senzorjev, bolj natančno bi lahko zgradili zemljevid, kar bi omogočalo bolj inteligenen način premikanja oziroma bi robot lahko bistveno hitreje preiskal celotni prostor kot pri naključnem načinu premikanja.

Kot smo ugotovili v tem diplomskem delu je problem inteligentnega premikanja po zunanjem prostoru precej kompleksen. Zgoraj opisane izboljšave predstavljajo le nekaj idej izmed večje množice potencialnih rešitev. Podjetja kot so iRobot, Samsung, Husqvarna in druga, ki se ukvarjajo s problemom inteligentnega premikanja robotov, v to vlagajo veliko denarja, časa in truda, da bi našli čim bolj optimalno rešitev. Naša rešitev ima torej še ogromno možnosti za nadaljnji razvoj, predstavlja pa že dokaj uspešno rešitev problema avtomatskega pometanja dvorišča.



# Literatura

- [1] Shimon Y. Nof, "Handbook of Industrial Robotics, Volume 1", 1999, str. 3-7
- [2] Zgodovina robotskih sesalnikov. Dostopno na:  
[http://en.wikipedia.org/wiki/Robotic\\_vacuum\\_cleaner](http://en.wikipedia.org/wiki/Robotic_vacuum_cleaner) (Oktober, 2014)
- [3] Delovanje bazne postaje. Dostopno na:  
[http://homesupport.irobot.com/app/answers/detail/a\\_id/491/~how-do-i-operate-the-roomba%C2%AE-home-base%3F](http://homesupport.irobot.com/app/answers/detail/a_id/491/~how-do-i-operate-the-roomba%C2%AE-home-base%3F) (Oktober, 2014)
- [4] Zgodovina podjetja iRobot. Dostopno na:  
[http://http://www.irobot.com/us/Company/About/Our\\_History.aspx](http://http://www.irobot.com/us/Company/About/Our_History.aspx) (Oktober, 2014)
- [5] Opis delovanja robota iRobot Roomba. Dostopno na:  
<http://www.sci.brooklyn.cuny.edu/~sklar/teaching/f06/cis1.0/papers/roomba-howstuffworks.pdf> (Oktober, 2014)
- [6] T. E. Kurt, "Hacking Roomba: ExtremeTech", 2007, str. 13-17
- [7] Lastnosti robota iRobot Roomba. Dostopno na:  
<http://www.irobot.com/us/learn/home/roomba.aspx> (Oktober, 2014)

- [8] Opis delovanja detekcije umazanije robota iRobot Roomba. Dostopno na:  
[http://homesupport.irobot.com/app/answers/detail/a\\_id/68/~/%3Fwhat-is-roombas-dirt-detect-technology%3F](http://homesupport.irobot.com/app/answers/detail/a_id/68/~/%3Fwhat-is-roombas-dirt-detect-technology%3F) (Oktober, 2014)
- [9] Opis senzorjev za detekcijo previsov robota iRobot Roomba. Dostopno na:  
[http://homesupport.irobot.com/app/answers/detail/a\\_id/314/~/%3Fwhat-are-cliff-sensors%3F](http://homesupport.irobot.com/app/answers/detail/a_id/314/~/%3Fwhat-are-cliff-sensors%3F) (Oktober, 2014)
- [10] G. Benet, F. Blanes, J. E. Simó, P. Pérez, "Using infrared sensors for distance measurement in mobile robots", *Robotics and autonomous systems*, št. 40, zv. 4, 2002
- [11] Opis delovanja odometričnega senzorja. Dostopno na:  
<http://www.robotappstore.com/Robopedia/Rotary-Encoder> (Oktober, 2014)
- [12] Opis funkcij robota Samsung NaviBot. Dostopno na:  
<http://www.samsung.com/au/consumer/home-appliances/vacuum-cleaner/robot-vacuum/VCR8980L4K/XSA-features> (Oktober, 2014)
- [13] Opis navigacijskega algoritma robota Samsung NaviBot. Dostopno na:  
<http://global.samsungtomorrow.com/?p=5271> (Oktober, 2014)
- [14] Prva robotska kosilnica MowBot. Dostopno na:  
<http://blog.modernmechanix.com/mowbot/> (Oktober, 2014)
- [15] Primer principa zakopane žice. Dostopno na:  
<http://www.bosch-indego.com/gb/en/howto/installation.html> (Oktober, 2014)
- [16] Opis delovanja principa zakopane žice na primeru nevidne pasje ograje. Dostopno na:

<http://science.howstuffworks.com/innovation/everyday-innovations/pet-fence6.htm> (Oktober, 2014)

- [17] Opis delovanja senzorjev za dež. Dostopno na:  
<http://www.doityourself.com/stry/how-does-a-rain-sensor-work>  
(Oktober, 2014)
- [18] Opis zgradbe in delovanja tipične robotske kosilnice. Dostopno na:  
<http://partners.nytimes.com/library/tech/00/05/circuits/articles/18howw.html> (Oktober, 2014)
- [19] Zgodovina podjetja Husqvarna. Dostopno na:  
<http://robomowerwiki.com/Automower> (Oktober, 2014)
- [20] Opis robotske kosilnice Husqvarna Automower 265 ACX. Dostopno na:  
<http://www.husqvarna.com/uk/products/robotic-mowers/automower-265-acx/> (Oktober, 2014)
- [21] Opis robotske kosilnice Husqvarna Automower Solar Hybrid. Dostopno na:  
<http://www.husqvarna.com/si/products/robotic-mowers/automower-solar-hybrid/> (Oktober, 2014)
- [22] Opis algoritma premikanja robotske kosilnice Husquvarna Automower. Stran 12. Dostopno na:  
<http://www.manualowl.com/m/Husqvarna/AUTOMOWER-230-ACX/Manual/329510> (Oktober, 2014)
- [23] Opis podjetja Karcher. Dostopno na:  
<http://en.wikipedia.org/wiki/K%C3%A4rcher> (Oktober, 2014)
- [24] Opis ročnega pometača Karcher S650. Dostopno na:  
[http://www.karcher.com/int/Products/Home\\_\\_Garden/Sweepers/Push\\_sweepers/17663000.htm](http://www.karcher.com/int/Products/Home__Garden/Sweepers/Push_sweepers/17663000.htm) (Oktober, 2014)

- 
- [25] Opis delovanja ročnega pometača. Dostopno na:  
<http://www.handkehrmaschine.info/kaercher-s-650/> (Oktober, 2014)
- [26] Specifikacije motorja Pololu. Dostopno na:  
<http://www.pololu.com/product/1443/specs> (Oktober, 2014)
- [27] Specifikacije vezja za krmiljenje motorjev RoboClaw. Dostopno na:  
<http://www.pololu.com/product/1492> (Oktober, 2014)
- [28] Opis delovanja ultrazvočnih senzorjev, ter primerjava s ostalimi podobnimi senzorji. Dostopno na:  
<http://www.generationrobots.com/en/content/65-ultrasonic-sonar-sensors-for-robots> (Oktober, 2014)
- [29] Platforma za razvoj prototipov Arduino. Dostopno na:  
<http://www.arduino.cc/> (Oktober, 2014)
- [30] C. Reas, B. Fry, "Getting Started with Processing", O'Reilly Media Inc., 2010
- [31] Primer razvojnega okolja za projekt Wiring. Dostopno na:  
<http://wiring.org.co/> (Oktober, 2014)
- [32] Opis razvojnega orodja Arduino IDE. Dostopno na:  
<http://en.wikipedia.org/wiki/Arduino#Software> (Oktober, 2014)
- [33] Opis razvojnega orodja Microsoft Visual Studio. Dostopno na:  
[http://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio) (Oktober, 2014)
- [34] Opis razširitve Visual Mikro. Dostopno na:  
<http://www.visualmicro.com/> (Oktober, 2014)
- [35] R. Kelly, V. S. Davila, J. A. L. Perez, "Control of Robot Manipulators in Joint Space", Springer Science & Business Media, 30. mar. 2006, str. 201-205

- [36] Opis algoritma proporcionalne regulacije za usklajevanja hitrosti dveh motorjev. Dostopno na:  
[http://www.robotc.net/wiki/Tutorials/Arduino\\_Projects/Mobile\\_Robotics/VEX/Using\\_encoders\\_to\\_drive\\_straight](http://www.robotc.net/wiki/Tutorials/Arduino_Projects/Mobile_Robotics/VEX/Using_encoders_to_drive_straight) (Oktober, 2014)